US006405245B1

(12) **United States Patent**
Burson et al.

(10) **Patent No.: US 6,405,245 B1**
(45) **Date of Patent: *Jun. 11, 2002**

(54) **SYSTEM AND METHOD FOR AUTOMATED ACCESS TO PERSONAL INFORMATION**

(75) Inventors: **Robert Burson; Dima Ulberg; Gregg Freishtat**, all of Atlanta, GA (US)

(73) Assignee: **Verticalone Corporation**, Atlanta, GA (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

This patent is subject to a terminal disclaimer.

(21) Appl. No.: **09/427,602**

(22) Filed: **Oct. 27, 1999**

Related U.S. Application Data

(60) Provisional application No. 60/105,917, filed on Oct. 28, 1998, and provisional application No. 60/134,395, filed on May 17, 1999.

(51) **Int. Cl.**[7] **G06F 13/00**
(52) **U.S. Cl.** **709/217**; 707/10; 345/705
(58) **Field of Search** 707/10; 709/230, 709/217; 345/705

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,347,632 A		9/1994	Filepp et al.	709/202
5,537,314 A		7/1996	Kanter	705/14
5,655,089 A		8/1997	Bucci	705/40
5,696,965 A	*	12/1997	Dedrick	707/10
5,699,528 A		12/1997	Hogan	705/40
5,710,887 A		1/1998	Chelliah et al.	705/26
5,712,979 A		1/1998	Graber et al.	709/224

(List continued on next page.)

FOREIGN PATENT DOCUMENTS

EP	0786728 A1	7/1997
EP	0848338 A	6/1998
WO	WO 98/28698	7/1998

OTHER PUBLICATIONS

"Collapsible User Interfaces for Information Retrieval Agents," Martin Frank and Pedro Szekely, Proceedings of International Conference on Intelligent User Interfaces, Jan. 5–8, 1999,Redondo, CA, pp. 15–22.

"A Softbot–based Interface to the Internet," Oren Etzioni and Daniel Weld, Communications of the ACM, vol. 37, No. 7, Jul. 1994, pp. 72–76.

"Strategic Directions in Database Systems—Breaking Out of the Box," *Avi Silberschatz and Stan Zdonik et al.*, ACM Computing Surveys, vol. 28, No. 4, pp. 764–778, Dec. (1996).

"Database Security and Privacy," Sushil Jajodia, ACM Computing Surveys, vol. 28, Issue 1 pp. 129–131, Mar. (1996).

"Managing Security and Privacy of Information," Sushil Jajodia, ACM Computing Surveys, vol. 28, Issue 4es, Dec. (1996).

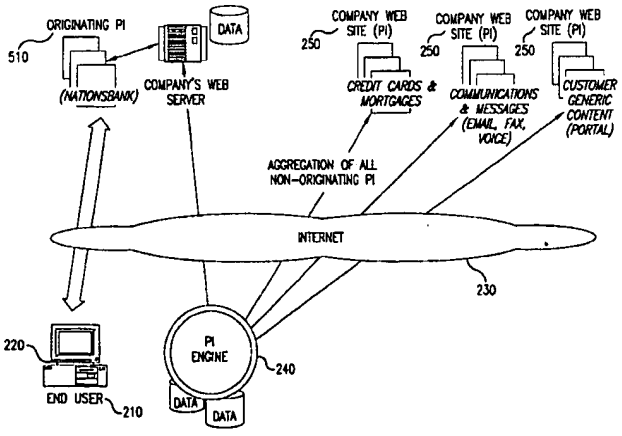
"Today's Style Sheet Standards: The Great Vision Blinded," Philip M. Marden, Jr. and Ethan V. Munson, IEEE Computer, pp. 123–125, Nov. (1999).

Primary Examiner—Kenneth R. Coulter
(74) *Attorney, Agent, or Firm*—Needle & Rosenberg, P.C.

(57) **ABSTRACT**

This invention is a system and method for automated access to personal information associated with an end user, wherein the personal information is stored on a personal information provider. A representation of the personal information and a link corresponding to the personal information stored on the personal information are presented to the end use via a client computer. Upon activation of the link, the client computer is automatically driven to the personal information provider presenting to the user via the client computer a page on the personal information provider.

9 Claims, 11 Drawing Sheets

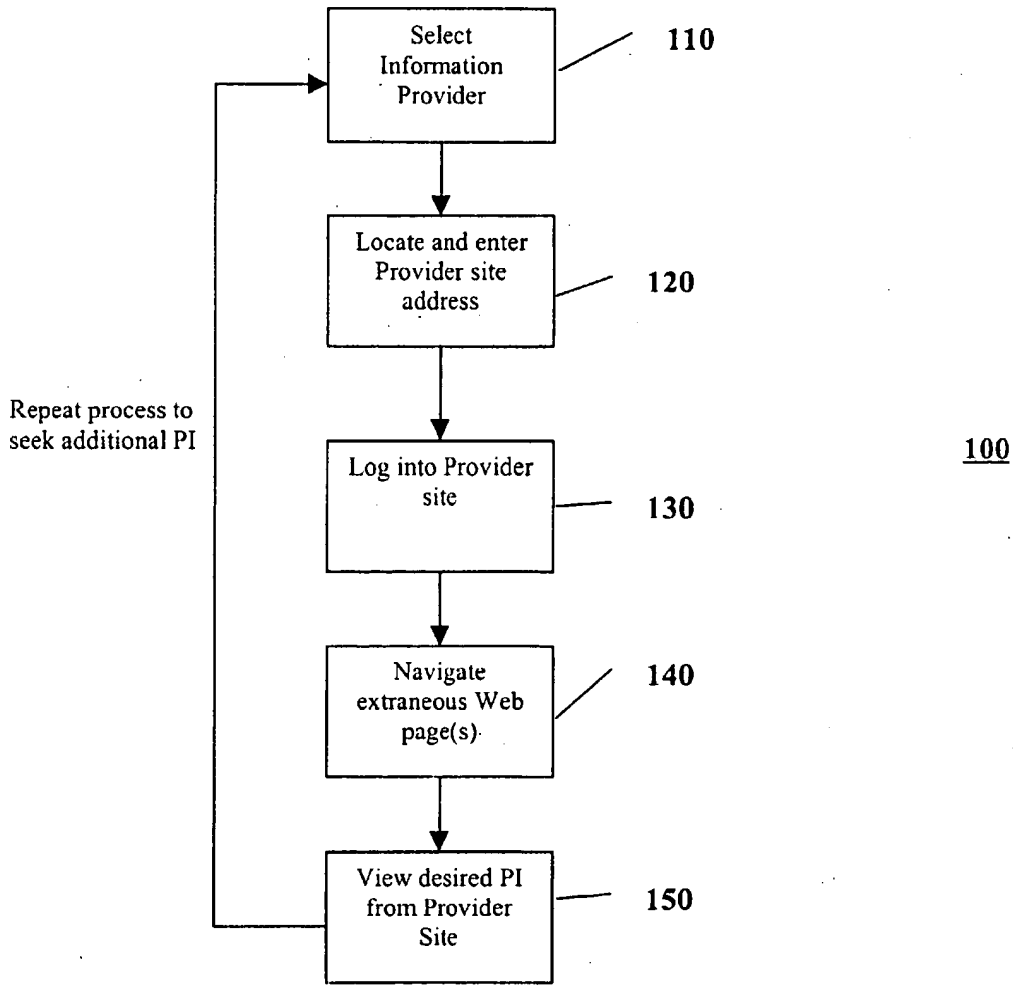


U.S. PATENT DOCUMENTS

5,724,567	A *	3/1998	Rose	707/2
5,825,884	A	10/1998	Zdepski et al.	705/78
5,848,396	A	12/1998	Gerace	705/10
5,860,068	A	1/1999	Cook	705/26
5,862,325	A *	1/1999	Reed et al.	709/201
5,878,219	A	3/1999	Vance, Jr. et al.	709/217
5,884,033	A	3/1999	Duvall et al.	709/206
5,884,045	A	3/1999	Kurihara	709/237
5,893,091	A	4/1999	Hunt et al.	707/3
5,894,554	A	4/1999	Lowery et al.	709/203
5,895,468	A	4/1999	Whitmyer, Jr.	707/10
5,897,622	A	4/1999	Blinn et al.	705/26
5,898,836	A	4/1999	Freivald et al.	709/218
5,913,202	A	6/1999	Motoyama	705/35
5,918,214	A	6/1999	Perkowski	705/27
5,926,798	A	7/1999	Carter	705/26
5,956,709	A	9/1999	Xue	707/3
5,963,915	A	10/1999	Kirsch	705/26
5,978,766	A	11/1999	Luciw	705/1
5,978,779	A *	11/1999	Stein et al.	705/37
5,983,200	A	11/1999	Slotznick	705/26
5,983,227	A	11/1999	Nazem et al.	707/10
5,987,440	A *	11/1999	O'Neil et al.	705/44
5,987,498	A	11/1999	Athing et al.	709/203
5,991,735	A	11/1999	Gerace	705/10
5,991,756	A	11/1999	Wu	707/3
5,995,965	A *	11/1999	Experton	707/10
6,006,227	A *	12/1999	Freeman et al.	707/7
6,029,175	A	2/2000	Chow et al.	707/104.1
6,317,783	B1 *	11/2001	Freishtat et al.	709/218

* cited by examiner

Figure 1
(Prior Art)



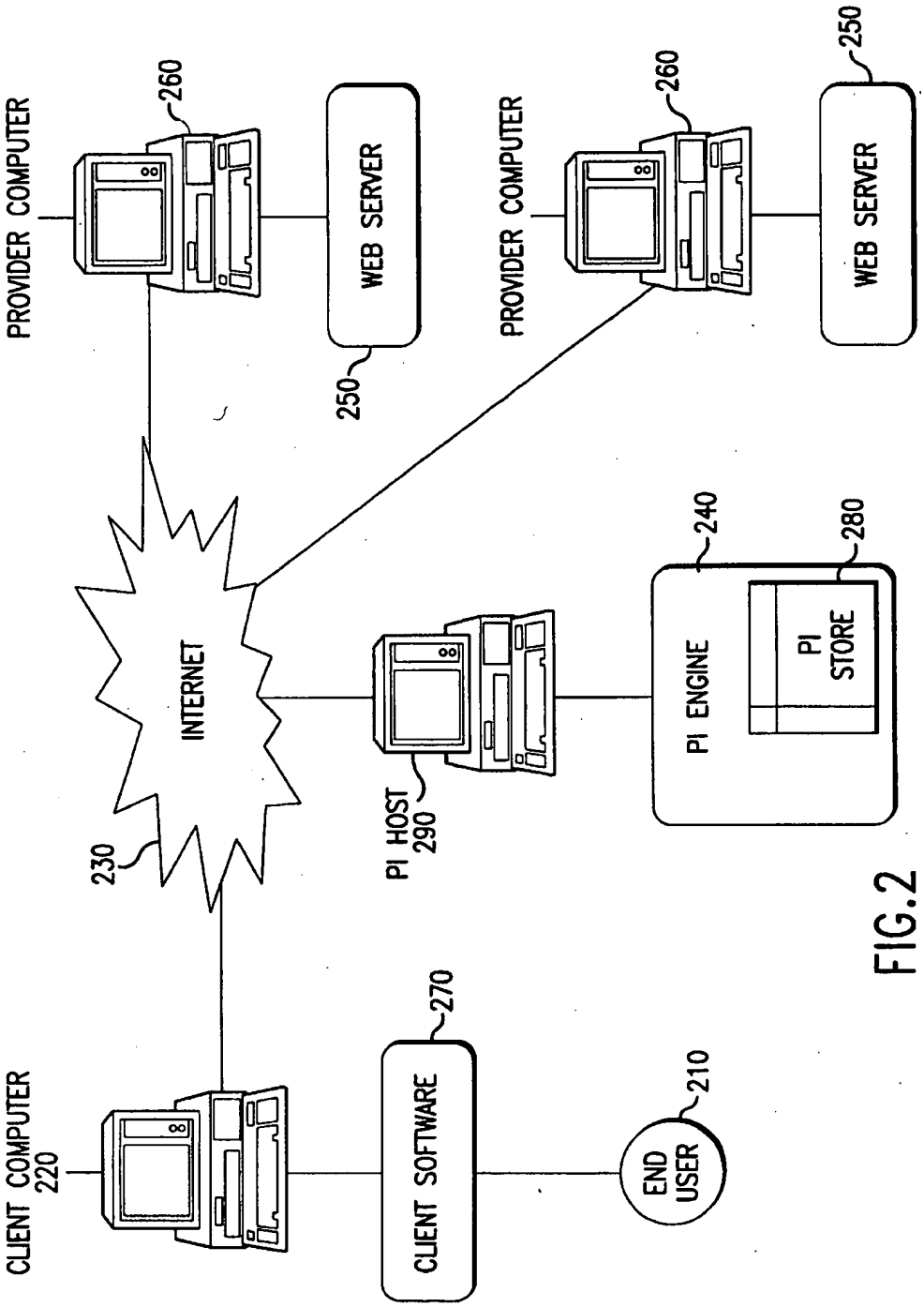

PROVIDER COMPUTER
260
WEB SERVER
250
PROVIDER COMPUTER
260
WEB SERVER
250
INTERNET
230
PI HOST
290
PI ENGINE
240
PI STORE
280
CLIENT COMPUTER
220
CLIENT SOFTWARE
270
END USER
210

FIG.2

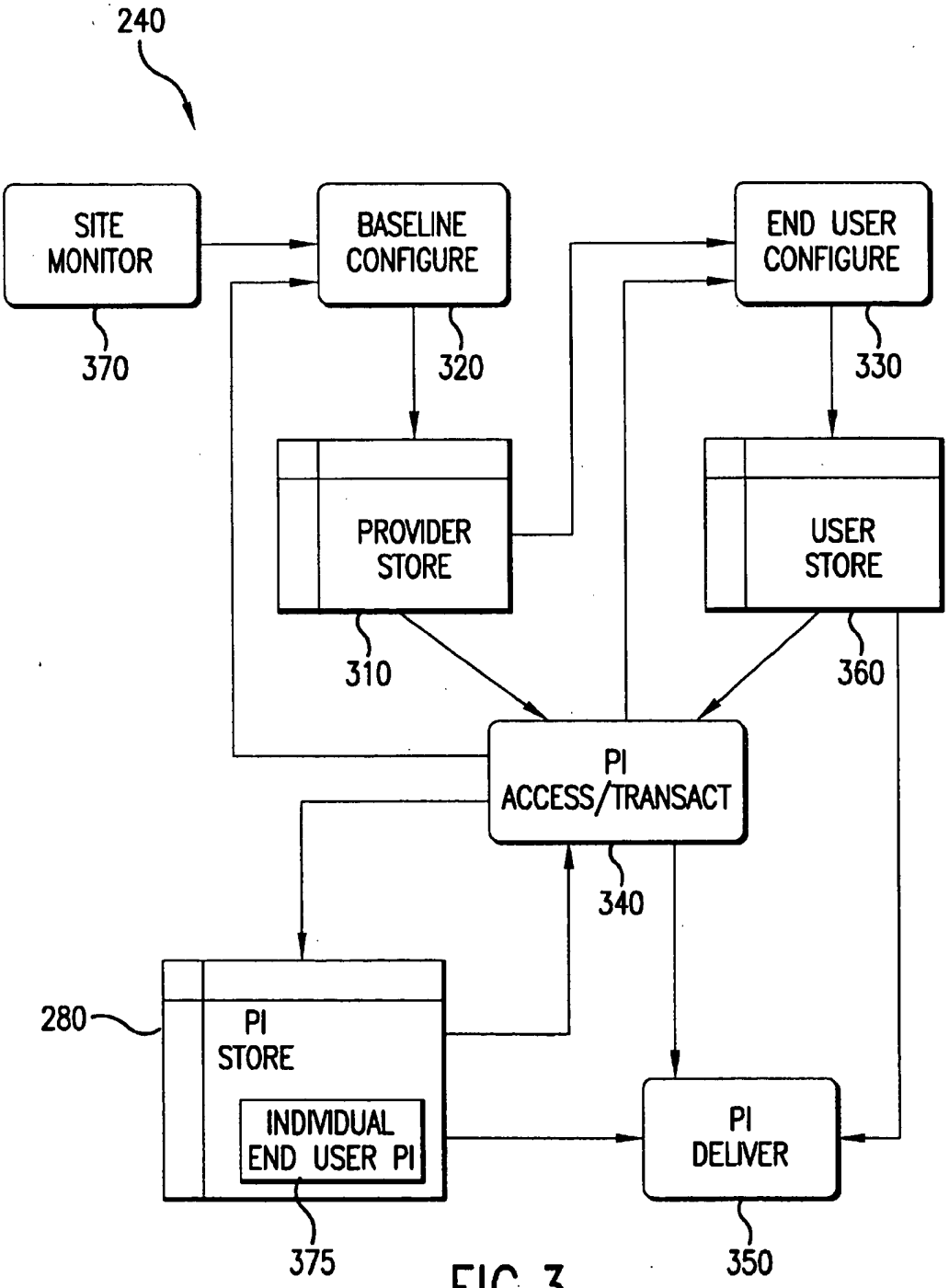

240
SITE MONITOR
370
BASELINE CONFIGURE
320
END USER CONFIGURE
330
PROVIDER STORE
310
USER STORE
360
PI ACCESS/TRANSACT
340
280
PI STORE
INDIVIDUAL END USER PI
375
PI DELIVER
350

FIG.3

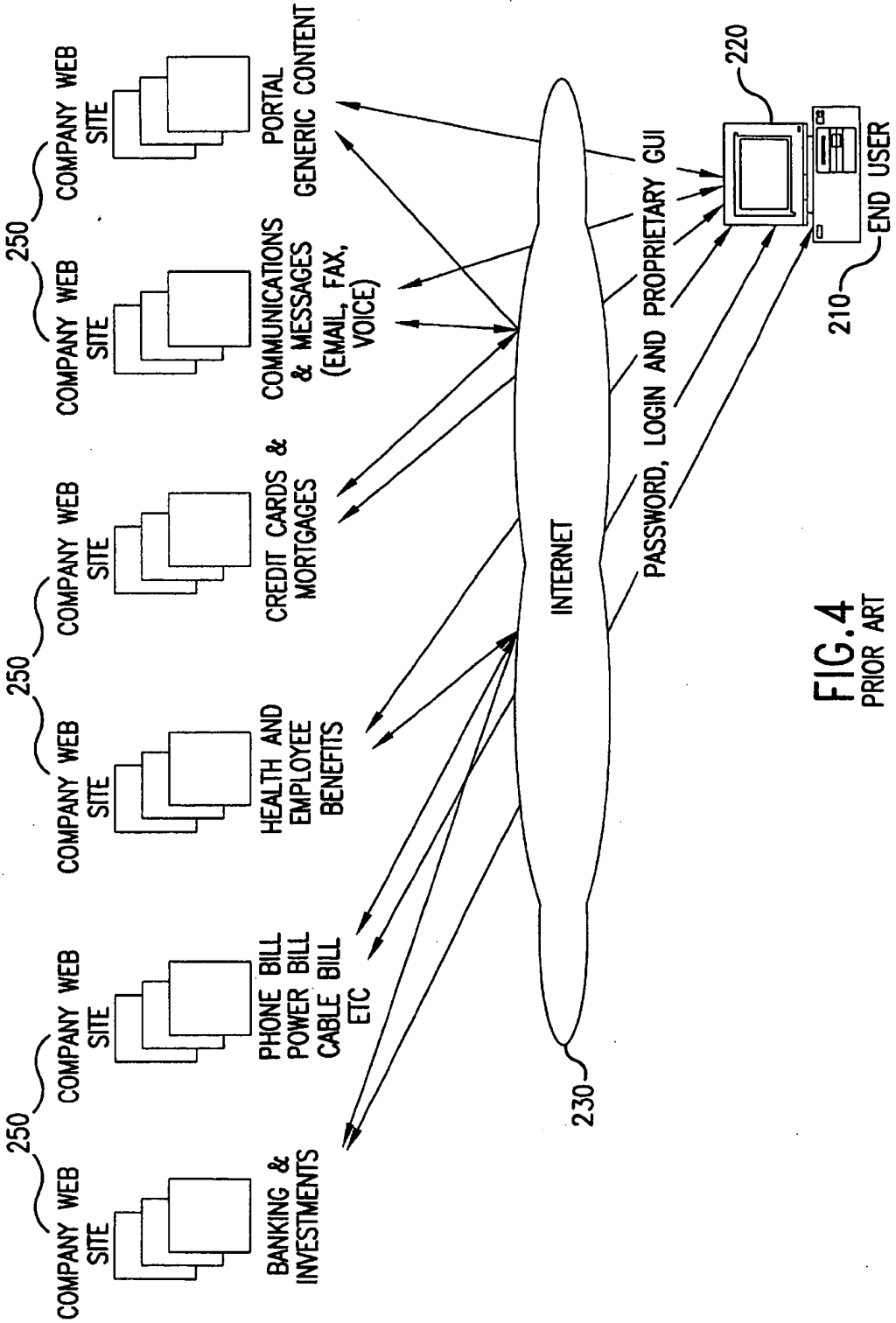


FIG.4
PRIOR ART

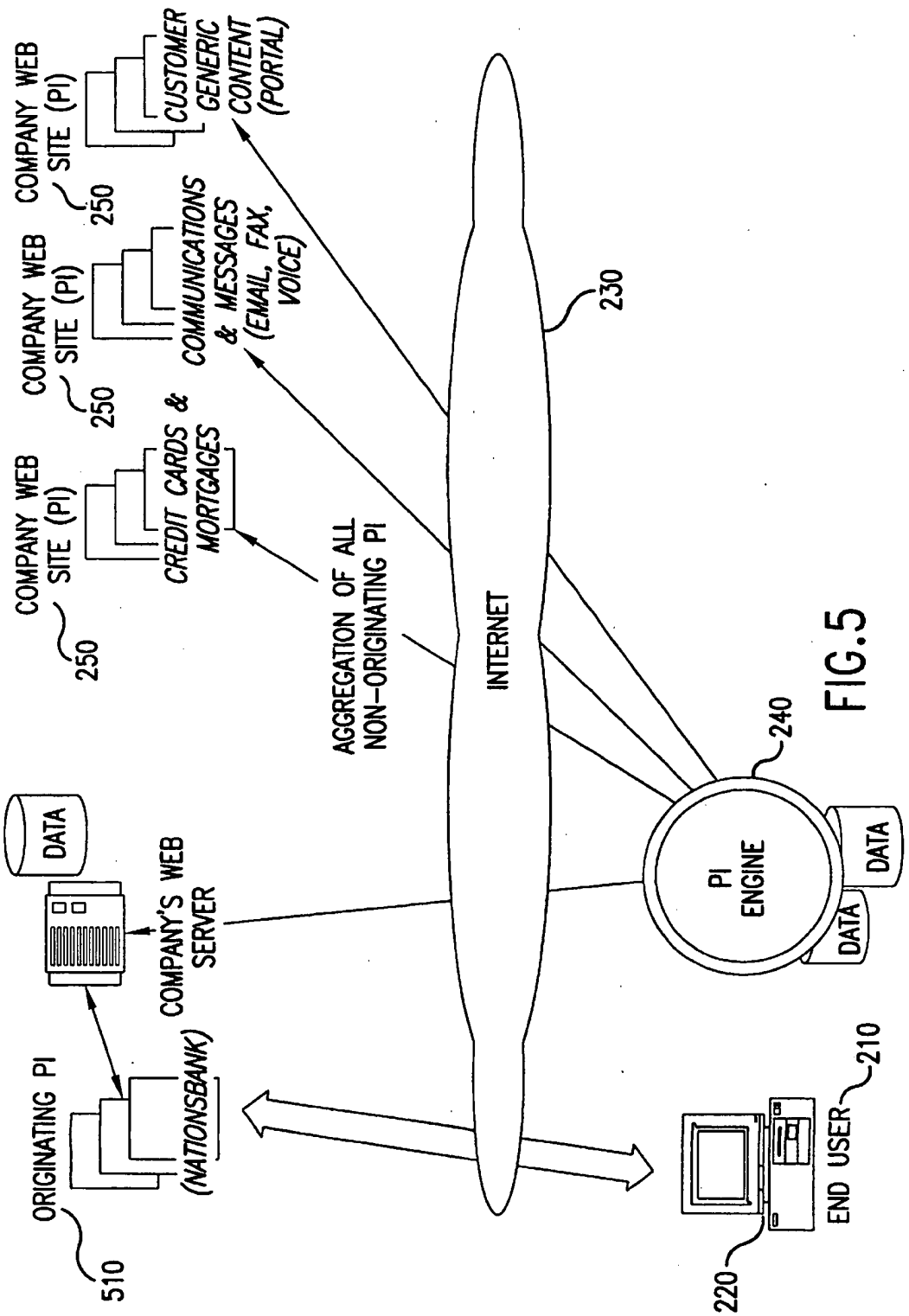
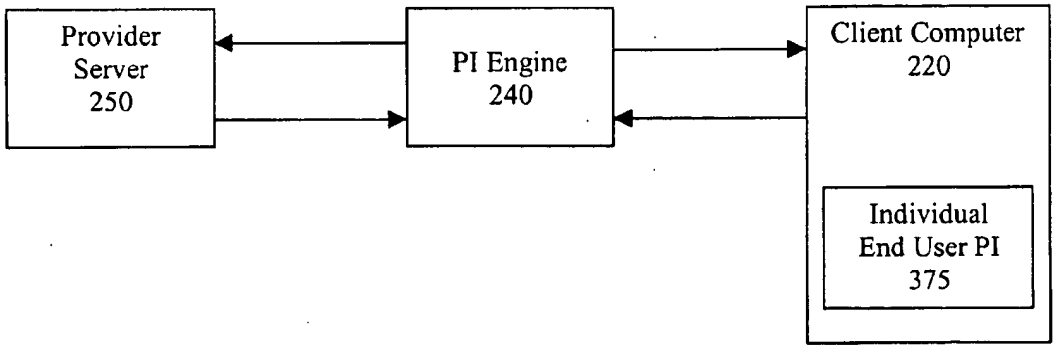

COMPANY WEB SITE (PI)
250
CUSTOMER GENERIC CONTENT (PORTAL)
COMPANY WEB SITE (PI)
250
COMMUNICATIONS & MESSAGES (EMAIL, FAX, VOICE)
COMPANY WEB SITE (PI)
250
CREDIT CARDS & MORTGAGES
AGGREGATION OF ALL NON-ORIGINATING PI
INTERNET
230
DATA
COMPANY'S WEB SERVER
ORIGINATING PI
(NATIONSBANK)
510
PI ENGINE
240
DATA
DATA
END USER
210
220

FIG.5

Figure 6


Provider
Server
250
PI Engine
240
Client Computer
220
Individual
End User PI
375

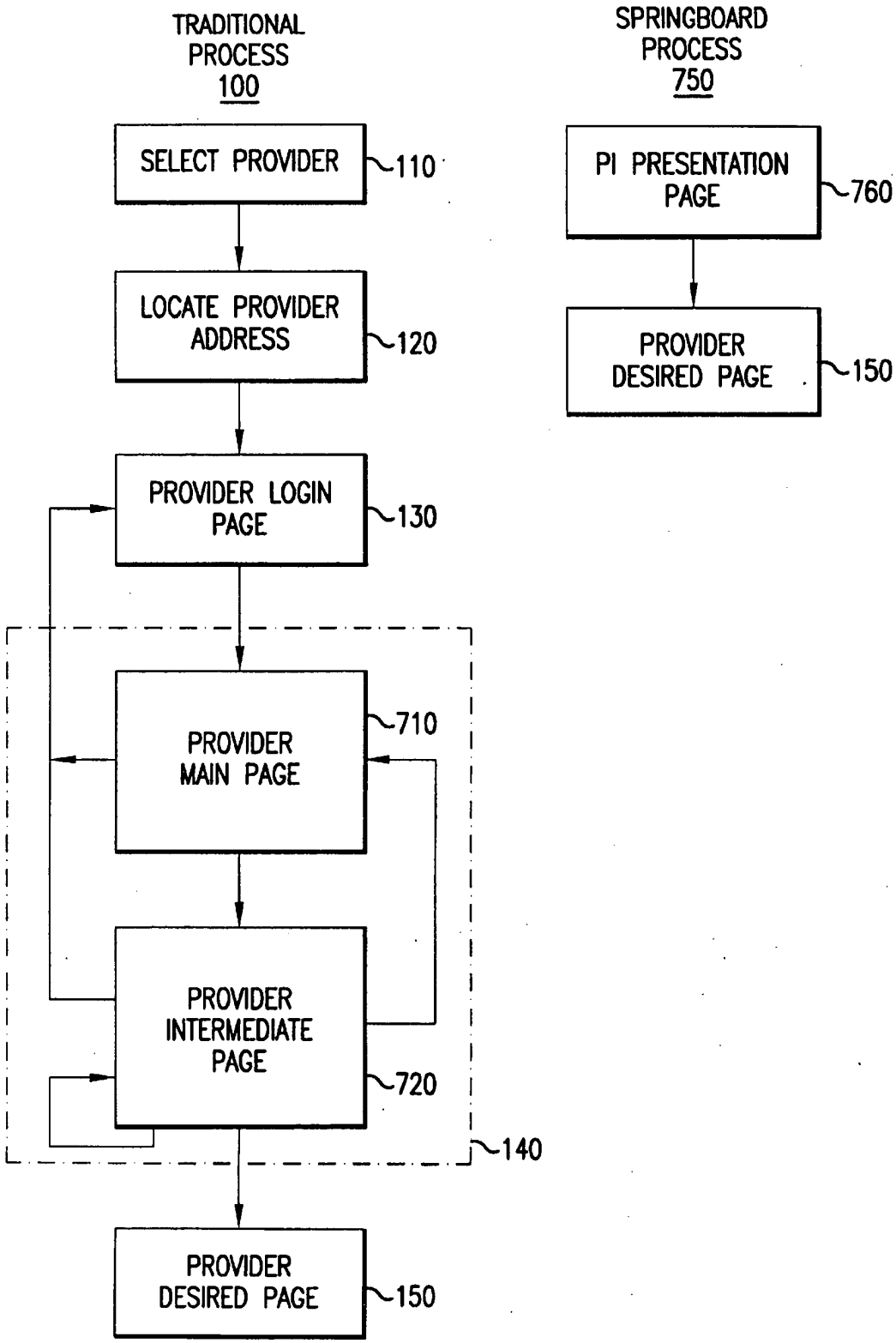

TRADITIONAL PROCESS 100
SELECT PROVIDER
110
LOCATE PROVIDER ADDRESS
120
PROVIDER LOGIN PAGE
130
PROVIDER MAIN PAGE
710
PROVIDER INTERMEDIATE PAGE
720
140
PROVIDER DESIRED PAGE
150
SPRINGBOARD PROCESS 750
PI PRESENTATION PAGE
760
PROVIDER DESIRED PAGE
150

FIG.7

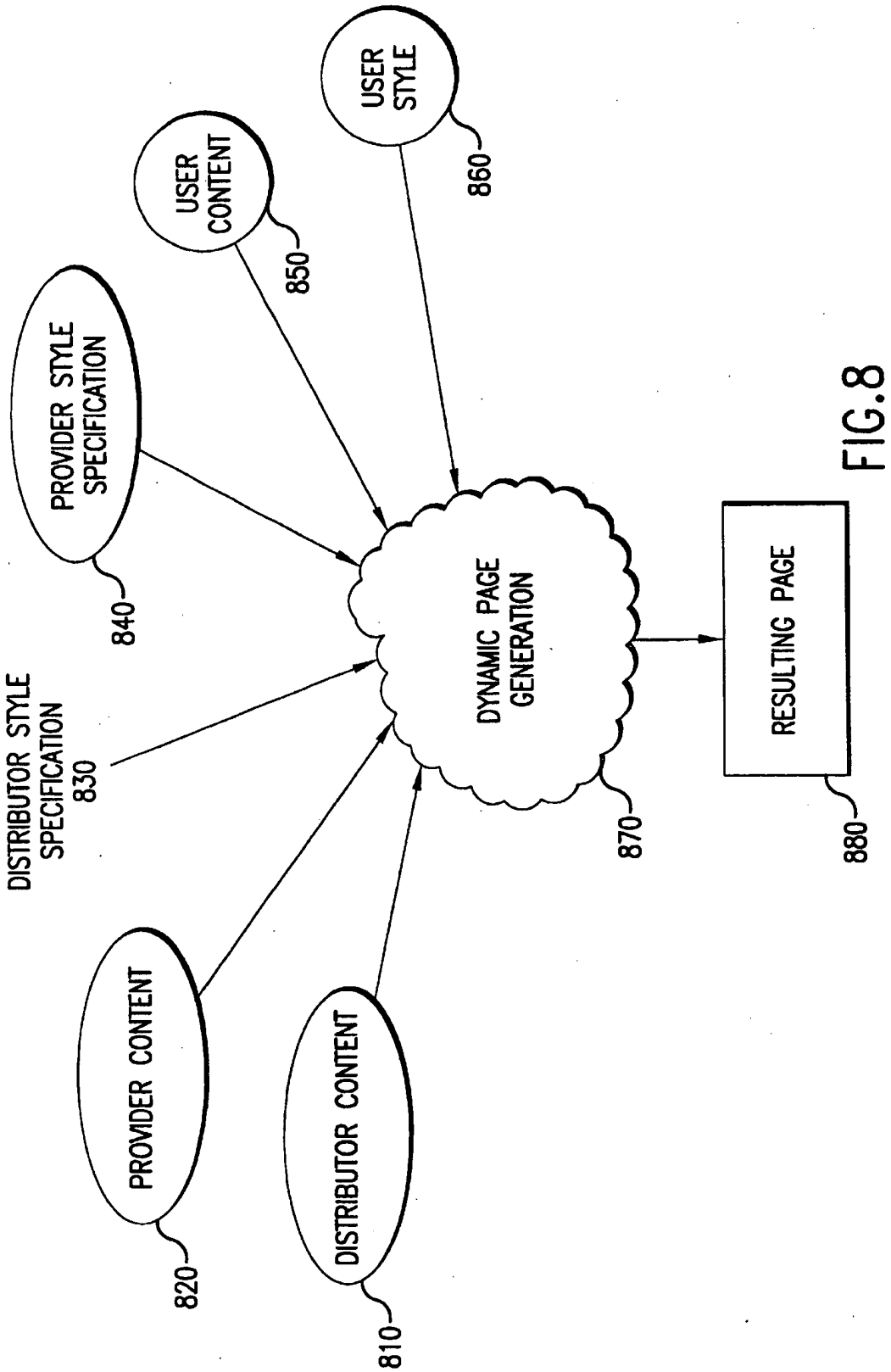

USER STYLE
860
USER CONTENT
850
PROVIDER STYLE SPECIFICATION
840
DISTRIBUTOR STYLE SPECIFICATION
830
PROVIDER CONTENT
820
DISTRIBUTOR CONTENT
810
DYNAMIC PAGE GENERATION
870
RESULTING PAGE
880

FIG.8

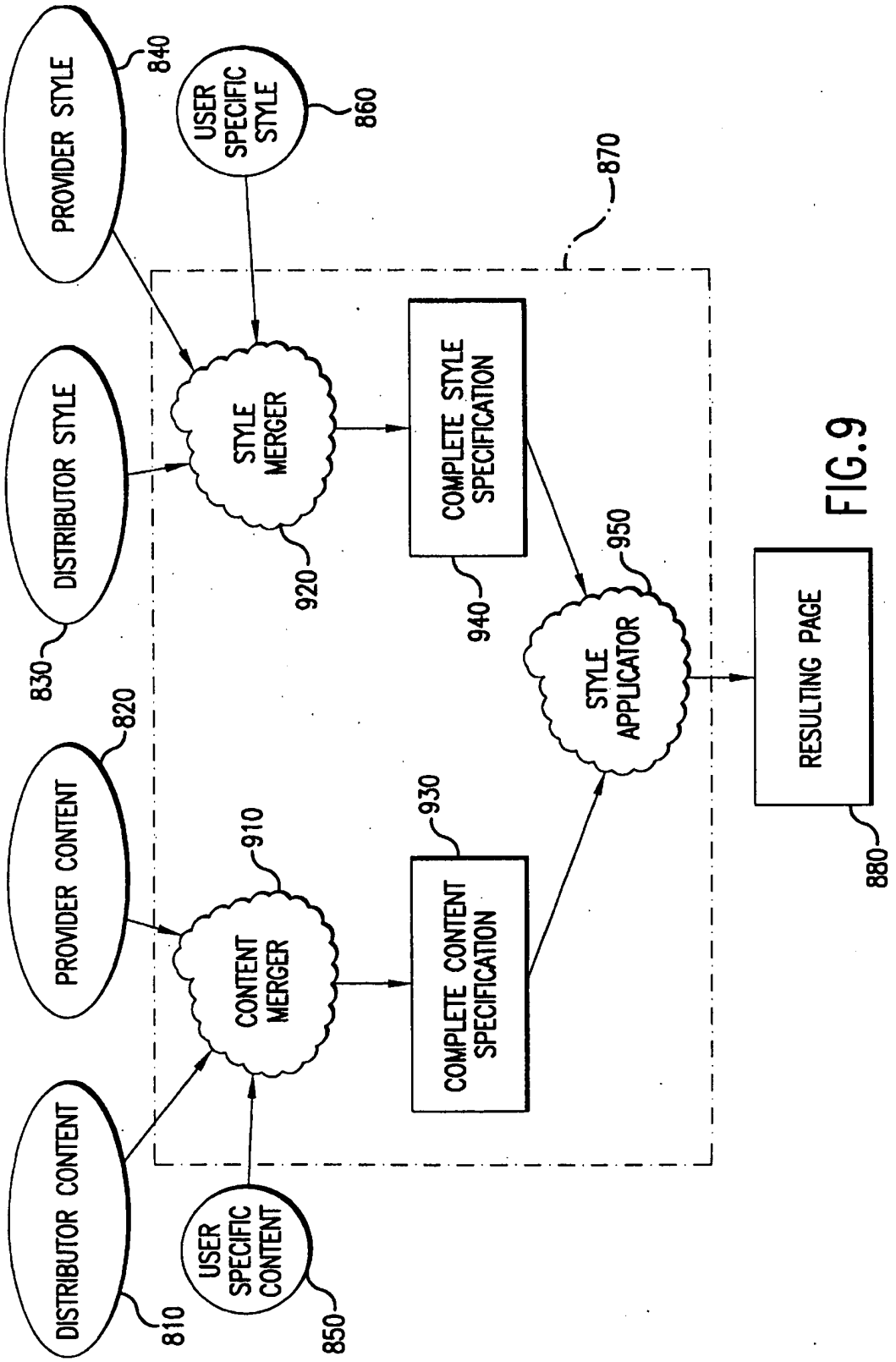

PROVIDER STYLE
840
USER SPECIFIC STYLE
860
870
DISTRIBUTOR STYLE
830
STYLE MERGER
920
COMPLETE STYLE SPECIFICATION
940
950
STYLE APPLICATOR
RESULTING PAGE
880
PROVIDER CONTENT
820
910
CONTENT MERGER
930
COMPLETE CONTENT SPECIFICATION
DISTRIBUTOR CONTENT
810
USER SPECIFIC CONTENT
850

FIG.9

Figure 10

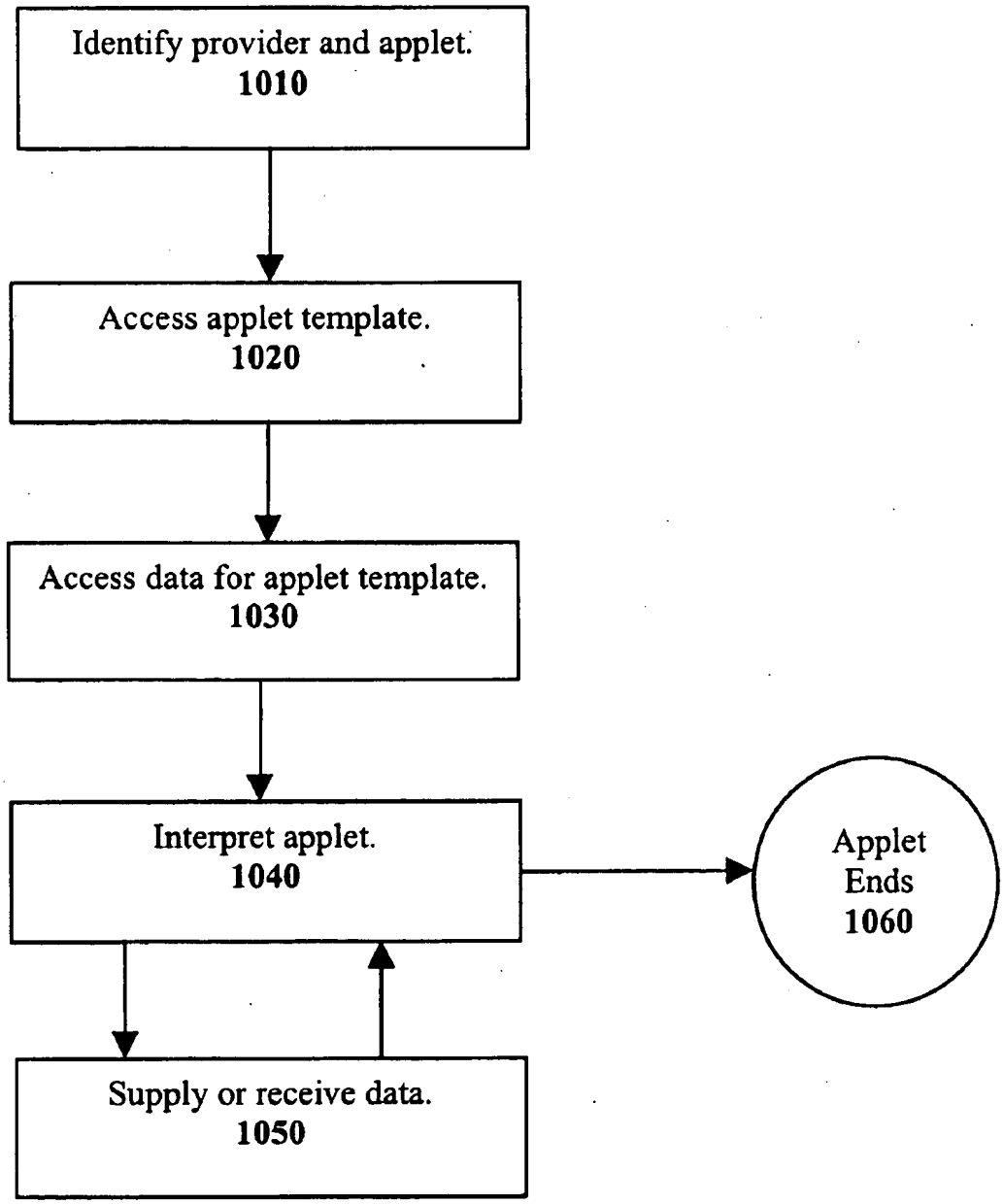
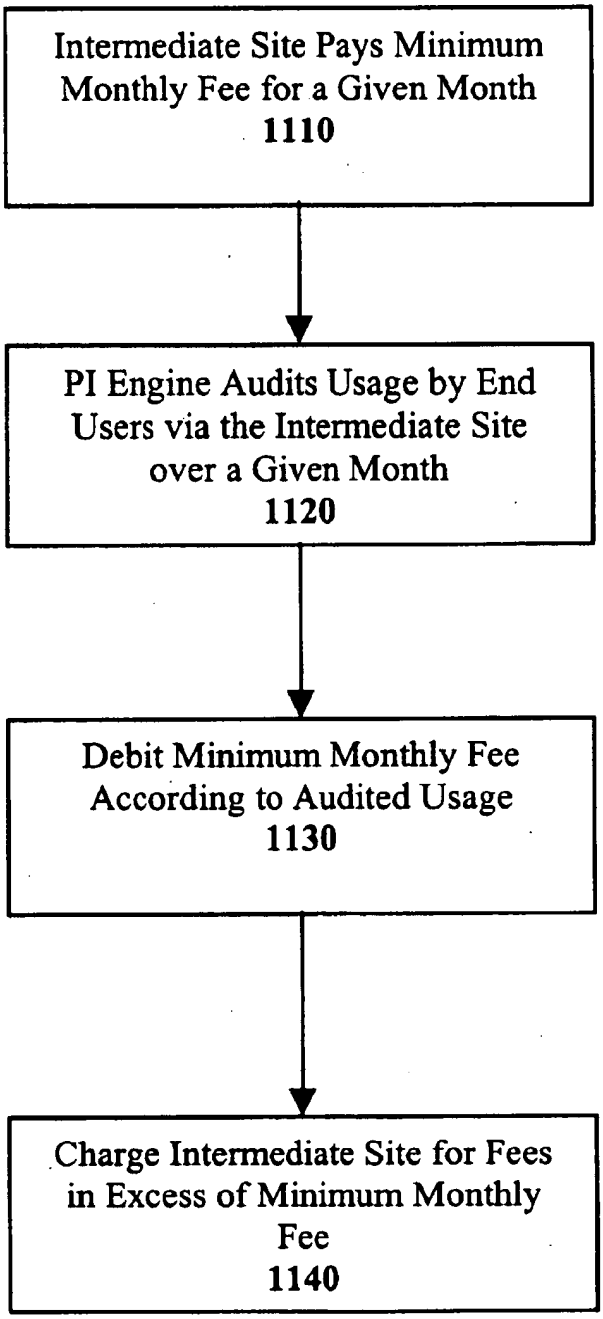

Identify provider and applet.
1010
Access applet template.
1020
Access data for applet template.
1030
Interpret applet.
1040
Applet
Ends
1060
Supply or receive data.
1050

Figure 11


Intermediate Site Pays Minimum Monthly Fee for a Given Month
1110
PI Engine Audits Usage by End Users via the Intermediate Site over a Given Month
1120
Debit Minimum Monthly Fee According to Audited Usage
1130
Charge Intermediate Site for Fees in Excess of Minimum Monthly Fee
1140

SYSTEM AND METHOD FOR AUTOMATED ACCESS TO PERSONAL INFORMATION

CROSS-REFERENCE TO RELATED PATENT APPLICATION

This application claims the benefit, pursuant to 35 U.S.C. §119(e), of applicants' provisional U.S. Patent Application Serial No. 60/105,917, filed Oct. 28, 1998, entitled "Apparatus and Method for Automated Aggregation and Delivery of and Transactions Involving Electronic Personal Information or Data" and of applicants' provisional U.S. Patent Application Serial No. 60/134,395, filed May 17, 1999, entitled "Apparatus and Method for Automated Aggregation and Delivery of and Transactions Involving Electronic Personal Information or Data".

BACKGROUND OF INVENTION

1. Field of Invention

The invention relates to a system and method for determining revenue derived from interactions involving personal information. The invention more particularly relates to the determination of revenue derived from an intermediary based on interactions involving personal information associated with end users aggregated from one or more personal information providers.

2. Description of Related Art

Looking back over the last five years, it is apparent that as the Internet gained momentum, consumers demanded applications or services that make their online experience simpler, easier to use, and more satisfying. The development of successful Internet Sites has corresponded with a number of themes which have developed over the last few years. When carefully analyzed this evolution is a logical development of the emerging digital economy.

Prior to 1994, the Internet was not a mass media, in part, because the existing technologies (FTP, Archie, Usenet, and Gopher) were not user friendly and required the end user to do all of the work (e.g., the end user had to learn of an existing data source, find the address, navigate to the destination, and download the information). As more consumers began accessing the Internet, Search Engines were created to solve this usability issue. With the advent of the commercial Search Engine, additional content could be easily added to the Internet and the end user had a means of finding and accessing this information. Consumers required better tools than Search Engines for organizing and accessing this wealth of generic content. Push technologies were explored, and eventually, the portal strategy was successfully adopted as an efficient way for consumers to easily access a variety of content sources in a single, easy to use format. As the volume of available online content continues to grow exponentially, portals are now confronted with the need to make different types of content available to different consumers based upon their particular preferences and tastes.

The phenomenal success of Internet portals and destination sites has demonstrated the importance of creatively and intelligently aggregating, organizing and presenting the mass of information available on the Web. Search engines, portals and destination sites have Internet strategies based on the frequency, duration and quality of end user visits to their sites. For this reason, destination sites and portals are constantly seeking content and/or technologies which drive quality traffic to their site and keep it there. Recent trends indicate that Internet users are up to 25 times more likely to

come back to a site when this information is organized according to personal preferences.

FIG. **1** displays the current process of acquiring online PI **100**. The end user first selects an information provider site in step **110**. The end user proceeds to step **120** by locating and entering the Internet address of the selected information provider. This step may be accomplished in several manners with varying levels of complexity. A simple means for accomplishing this step is the utilization of a bookmark or favorite whereas locating an information provider for the first time might involve significant time and effort performing online searches. In step **130**, the end users logs into the selected information provider's Web site utilizing the site's specific logon protocol. This protocol usually involves verifying the identity of the end user using a user name and password or other means of verification, acquiring the verification data from cookies residing on the end user's system or a combination of requested data and cookie data. The end user continues in step **140** by navigating through Web pages on the information provider's Web site until the desired information is located. During this process, the end user is often required to visit Web pages of little or no use to the end user whose goals is to simply acquire the particular PI residing on the Web site. Ultimately in step **150**, the end user is presented with the desired PI. The entire process **100** is repeated for each individual piece of PI desired by the end user. Under this PI access model, the end user must visit each separate information provider, track potentially different identity verification data for each, utilize a different user interface at each site and possibly wade through a significant number of filler Web pages.

FIG. **4** pictorial illustrates the architecture of this current access process. The end user **210** utilizes the client computer **220** to access each PI Web site **250** across the Internet **230**. This current model suffers from several significant deficiencies. The end user must login to each site separately. Each separate site has its own graphical user interface. Each site wants the end user to stay and return; each visited site wants to retain end user focus for as long as possible. No true aggregation of PI exists; multiple accesses simply allow sequential access to particular pieces of PI.

One partial solution to these problems has recently evolved in the form of portal sites. Generic portal sites aggregate resources into categories and provide links to sites covering topics within those categories. Yahoo and Excite are examples of generic portal sites. These sites facilitate horizontal aggregation of generic content; horizontal aggregation refers to aggregation of PI access within a particular information provider category such as banks or utility companies. Some portal site allows individual end users a limited capability to select and configure disparate generic PI. Generic PI refers to PI of interest to the particular end user that does not require specific identity verification to obtain. For example, an end user might be interested in the weather forecast for his local area. This information could be integrated into a portal page without requiring identity verification of the particular end user receiving this PI. The individualized portal page provides a significant benefit to users seeking to aggregate generic PI. However, current portal pages do not generally provide PI requiring identity verification such as an end user's stock portfolio or bank balance. Further, these pages do not facilitate transactions utilizing PI.

Under current technology, aggregating PI available over the Internet requires a significant burden in terms of time, effort and learning curve. An end user wishing to access his PI needs to individually visit a variety of information

provider sites each with its own requirements, graphical user interface and login protocol.

SUMMARY OF THE INVENTION

The present invention is a system and method for automated access to personal information associated with an end user, wherein the personal information is stored on a personal information provider. A representation of the personal information and a link corresponding to the personal information stored on the personal information are presented to the end user via a client computer. Upon activation of the link, the client computer is automatically driven to the personal information provider presenting to the user via the client computer a page on the personal information provider.

In one embodiment, an application is downloaded to the client. The downloaded application initiates a connection between the client computer and the personal information provider. The application navigates through pages on the personal information provider until arriving at the personal information. Finally, the application presents the personal information to the user on the client computer. The application may be either generated with any necessary data associated with the end user and associated with the personal information or such data may be transmitted to the application. The data associated with the personal information provider may include a navigation script for guiding the application to the personal information. The data associated with the end user may include any data necessary to effectuate the navigation via the navigation script.

In a further embodiment, a message including any necessary user data and personal information provider data is transmitted to the client computer causing the client computer to automatically log the end user into the personal information provider, thereby leaving the end user at a post login page. In a preferred embodiment, the message comprises a page containing a form, which includes login information that upon opening by software on the client computer redirects the client computer to a post login page.

In yet a further embodiment, the client computer is driven to the personal information by connecting to the personal information provider, navigating to the personal information on the personal information provider, presenting the personal information to the end user via the client computer and proxying subsequent interactions between the client computer and the personal information provider for a given session with the personal information provider.

The above and other objects and advantages of the present invention will become more readily apparent when reference is made to the following description, taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is a process diagram of the current process that end users perform to access Internet available PI.

FIG. **2** is a block diagram of the components that could be used to implement present invention.

FIG. **3** is a block diagram of the components of the PI engine.

FIG. **4** is a diagram of the current PI access architecture.

FIG. **5** is a diagram of an architecture supporting PI access utilizing an intermediary Web site.

FIG. **6** is a diagram of the cookie/client cache architecture.

FIG. **7** is a flowchart for accessing pages underlying particular PI via the traditional process of FIG. **1** and via springboard technology.

FIG. **8** depicts the integration model for the dynamic generation of HTML pages.

FIG. **9** displays the run-time process for dynamic generation of HTML page.

FIG. **10** illustrates a process for automated applet interaction utilizing a modified Java virtual machine.

FIG. **11** is a flowchart exemplifying an intermediary Web site transaction structure.

DETAILED DESCRIPTION OF THE INVENTION

A preferred embodiment of the invention is now described in detail. Referring to the drawings, like numbers indicate like parts throughout the views. As used in the description herein and throughout the claims that follow, the meaning of "a," "an," and "the" includes plural reference unless the context clearly dictates otherwise. Also, as used in the description herein and throughout the claims that follow, the meaning of "in" includes "in" and "on" unless the context clearly dictates otherwise.

In no time, end users will have to log into a large number of different Web Sites, each with separate passwords, security, rules, software and "look and feel"—just to get the information currently obtained by checking one place—the mailbox at the end of the driveway. The Internet will fundamentally change the way in which end users will access Personal Information (PI) and will make e-commerce as familiar as using an ATM. "Personal Information" is all of the data that companies, information providers, have that is specific or unique to each person such as monthly bills, bank account balances, investments information, health care benefits, email, voice and fax messages, **401**(*k*) holdings or potentially any other information pertinent to a particular end user.

The present invention alleviates several of the problems with the current PI acquisition methods by automatically aggregating PI, not only generic PI as aggregated by portals but also PI specific to the end user requiring identity verification for access. In one embodiment, the invention automates the PI acquisition and delivery process. FIG. **2** provides a block diagram of components that could be used to implement the present invention. The end user **210** accesses a client computer **220** running client software **270** which in a particular embodiment could be a general Web browser such as Navigator or Communicator (Netscape). The client computer **220** utilizes the Internet **230** to access a PI engine **240** running on a PI host **290**. The PI engine **240** examines stored PI **280** for freshness. Any stale PI items are refreshed by directly reacquiring the PI from the particular information provider's Web site **250** running on the provider's computer system **260** accessed across the Internet **230**. The PI engine **240** stores the fresh PI in its store **280** and delivers the PI to a selected destination, in this instance across the Internet **230** to the client computer **220** which displays the information to the end user **210** using the client software **270**. The PI engine **240** refreshes all stale PI in a like manner prior to forwarding the aggregated PI to both the store **280** and the delivery destination, the client computer **220** in this instance. The PI engine **240** may refresh the PI sequentially or in parallel. For example, the end user's checking account balance would be updated through his bank's Web site, his email from his particular email site, his portfolio information from his broker's site and his electricity bill from his electricity company's site.

FIG. **3** displays a block diagram of the components of the PI engine **240**. The PI engine **240** is composed of both

storage and processing components. The three primary storage components are the PI store **280**, the PI Provider store **310** and the user store **360**. The first storage component of the PI engine **240** is the PI store **280**. The PI store **280** contains each individual's PI record **375**; the PI associated with a particular end user is segregated from the PI of all other end users. The PI engine also utilizes a provider store **310** that maintains general parameters associated with particular PI providers. The general parameters of a PI provider define the types of verification data necessary and the procedures to be followed to gain access to the particular PI provider. Each PI provider record also contains the types of PI provided by the PI provider and the types of transactions supported by the provider. Along with the type of PI or transaction, the record also contains the additional types of data and procedures necessary to access the PI or execute the transaction. A user store **360** is also necessary to maintain configuration and verification information concerning particular end users. For each end user, the user selected PI providers, PI and transactions are registered along with the verification data necessary to acquire the PI or execute the transaction from the PI provider.

The PI store **280** may be implemented in a variety of ways. Referring to FIG. **2**, the PI store **280** may comprise a database residing on the PI Host **290**. Under this approach, the PI for each individual end user **210** is stored as a separate record or object **375** in the database. In yet another embodiment, the PI for each end user **210** could be stored in a separate file **375**, thus performing the task of segregating PI of different users at the file level.

In addition, or as an alternative, the PI associated with each end user **210** may reside on his/her client computer **220** using cookie technology as specified in D. Kristol and L. Montulli, "HTTP State Management Mechanism", Request For Comments (RFC) 2109, February, 1997 (available at http://www.ietf.org/rfc/rfc2109.txt), which is expressly incorporated herein in its entirety. The PI associate with the end user **210** would be stored as PI cookies **375**. This implementation mechanism provides inherent support for segregating PI associated with one end user **375** from PI associated with all other end users. Utilizing this method as a substitute for a centralized store provides a layer of security against unauthorized access. As a further measure, PI data stored in cookies could be stored in an encrypted format.

FIG. **6** provides a diagram of a typical implementation of the PI store **280** using cookie technology; references in the foregoing description are also made to FIG. **3** with respect to the internal workings of the PI engine **240**. When an attempt is made to access PI by an end user **210** directly, or through an intermediary Web server, the PI access/transact component **340** of the PI engine **240** would retrieve stored PI **375** from the PI store **280**. Under this approach, this stored PI **375** would be received directly from cookies sent by the client computer **220** of the end user **210**. The PI access/transact component **340** would perform any decryption if necessary. Any updates required would be obtained by direct access of PI providers **250**. The PI deliver component **350** would provide the mechanism for both updating the PI store **280** as well as transmitting the requested PI to the end user **210**, directly or through an intermediary Web site. The PI deliver component **350** would place the updated PI in the PI store **280** by replacing the outdated PI cookies **375** stored on the client computer **220**. The PI deliver component **350** would also handle any encryption if necessary. The PI deliver component **350** would also be responsible for transmitting requested PI. In a preferred embodiment, the PI store **280** would be implemented using this cookie-based architecture.

The user store **360** may be implemented in a variety of ways. Referring to FIG. **2**, the user store **360** may comprise a database residing on the PI Host **290**. Under this approach, the personal configuration data for each individual end user **210** is stored as a separate record or object in the database. In addition, or as an alternative, the end user data could be distributed in a manner similar to the cookie/cache architecture describe above with respect to the PI store **280**.

In a preferred embodiment, the user store **360** could be implemented through personal information configuration (PIC) files. PIC files store a personal profile such as name, address, and social security number in secure, encrypted fashion for each end user. PIC files facilitate automatic registration of end users with information Providers via the end user configuration component **330**. This component will read the PIC file and, using retrieved personal information, pre-populate registration templates for selected Providers. Then, it will prompt the user to enter required information that is missing from profile, if necessary. If the information is complete, the registration is automatically completed. Next, the end user configure component **330** completes any Provider registration forms, gets responses and updates the end user's PIC.

The four primary processing components access and manipulate the data in the three stores. The processing components may execute on a single processor, such as a file server computer system based on a Pentium class (MMX, PRO, II, III, etc.) central processing unit or an equivalent, or multiple processors. These four processing components are the Baseline configure component **320**, the end user configure component **330**, the PI access/transact component **340** and the PI delivery component **350** as seen in FIG. **3**. The Baseline configure component **320** provides the interface by which new user selectable PI providers are added to the system. This component **320** might be implemented in a variety of ways including trial and error followed by manual entry of configuration information, semi-automated trial and error (automated location of Hypertext Markup Language (HTML) <FORM> elements, Javascript functions and Java applets) followed by manual entry of configuration information or, preferably, configuration by example (executing the protocol in a simulated Web client where the simulated Web client automatically generates a list of required data and a list of steps in the access process). These processes would be utilized at two levels: the first level being the set of data and steps required for general access to the particular PI provider and the second level being the set of additional data and steps required for accessing each particular piece of PI on the PI provider's site. The baseline configuration component **320** may be triggered independently when a new PI provider is added to the system, or it might be triggered as a result of a failure of the PI access/transact component **340** potentially indicating a change in access requirements for the failed access. This latter warning would more likely result where the PI access/transact component **340** has made a comparison between requirements supplied by the Provider store **310**, both general to the PI provider and specific to the PI or transaction, and the end user data supplied by the user store **360** after seeking end user verification via a request of the end user to confirm the previously entered required access data via the end user configure component **330** and found an inconsistency. When an inconsistency is determined, updates to the Provider store **320** are made to bring the Provider data into conformance with current access/transaction requirements.

The end user configure component **330** allows an end user to select and configure PI and transactions of interest to the

specific user. This configuration information is maintained in the user store **360**. When an end user initially subscribes to the system according to the present invention, the system allows the user to select the types and sources of PI and/or transactions desired. First, the system requests permission from the end user to act on his behalf to obtain any selected PI and to execute any authorized transactions. Next, the system provides the user with a list of known information suppliers and the types of PI supplied from and transactions supported by the particular PI provider from the Provider store **320**. The system requests the verification data necessary for accessing each selected PI provider and the additional data required by the particular PIs and/or transactions desired from that PI provider. Assuming the end user is already a registered user with the selected PI provider or the particular PI provider does not require prior registration, the data supplied by the end user is placed in the user store **360**.

One method of obtaining any cookie data would be for the end user to access each previously accessed PI utilizing the PI engine **240** as a proxy server. The PI engine **240** would pass the cookie data to the PI provider site with the appropriate Web page requests to obtain the PI or execute the transaction and with the end user's permission retain a copy of the cookie data in the his record in the user store **360**. An alternate means of obtaining the cookie data would be a direct upload of the cookie information from the end user's computer. In a preferred embodiment, no cookie data is necessary where a user is already registered with a provider. All that is necessary is the verification data for login.

If the end user does not have the requisite information because he is not a registered user of a selected PI provider, the user configure component **330** prompts the user for the information necessary to register the end user with the PI provider and performs the registration procedure required by the PI provider. A simulated Web client could perform this process automatically supplying the access data as required and sending any necessary cookie data. The manner in which such a simulated client registers the end user depends significantly upon the interaction method used on the PI provider Web site. If the Web site uses HTML forms and common gateway interface (CGI) applications, the end user configure component **330** can formulate a uniform resource locator (URL) to replicate the effect of actual form usage and submit this URL to the simulated Web client. The use of a URL to mimic an HTML form is equivalent to manually entering the data into the Web <FORM> element. See Kerven, Foust, Zakour, *HTML* 3.2 *Plus How-To*, Waite Group Press, 1997, pp. 559–569. If the Web site uses a mixture of HTML forms and Javascript functions, a simulated Web client with a modified Javascript interpreter could effectively register the user by following the end user registration process for the particular PI provider. The registration process to follow would be obtained from the record of the particular PI provider in the Provider store **320**. The Javascript interpreter in the simulated Web client would follow this procedure and supply the data supplied by the end user. A similar process could be used if the registration process on the PI provider Web site utilizes a Java applet. A Web client with a modified Java bytecode interpreter could effectively register the user by following the end user registration process stored for the particular PI provider in the Provider store **320**. The bytecode interpreter would supply the data previously entered by the end user rather than requiring interactive input from the end user. If the PI provider Web site utilizes a combination of forms, scripts and applets, the individual procedures above could be used in combination to accomplish the desired registration.

With reference to FIG. **2** and FIG. **3**, a modification of the Java virtual machine (VM) could allow for automated interaction between the various functional components of the PI Engine **240** and Java applet available through provider Web servers **250**. Templates for interacting with particular applets could reside in the Provider store **310**. The specific input data utilized by such templates could be stored in the User store **360**. When a functional component such as the end user configure **330** or the access/transact **340** components requires automated communication with a Java applet on a provider Web server **250**, the modified Java VM would facilitate this interaction.

FIG. **10** illustrates one process utilizing such a modified Java VM to achieve such automated interaction. The functional component requiring interaction identifies the provider and the particular applet on that provider with which the component needs to interact in step **1010**. In step **1020**, the component accesses the necessary template for interacting with the applet from the Provider store **310**. Proceeding to step **1030**, the component accesses the User store **360** to obtain the data required by the template. The modified Java VM interprets the applet in step **1040** and, rather than requiring interactive input from a user as in a normal Java applet execution, awaits input from or output to the interacting functional component of the PI engine. In step **1050**, the functional component supplies input data to the modified Java VM according to the accessed template and retrieved data and receives output data according to the accessed template. Steps **1040** and **1050** repeat so long as additional input to or output from the applet continues. Upon termination of the applet, the functional component continues with its own processing in step **1060**.

A successful registration could result in displaying the registration information to the end user for future reference. Further, the end user configure component **330** stores the requisite access verification data for the PI provider and the additional data required to access the selected PI or transaction in the user store **360**.

In a preferred embodiment of such automated registration, any necessary cookie data would be accepted and stored as needed by the end user configure component **330**. In many cases, cookie data is session specific and, therefore, of little long term utility. Cookies generated during the registration process are used solely during the registration process then discarded once registration is complete.

A failed registration could result from several situations. First, the end user attempting to register with the PI provider does not qualify for registration; for example, an end user attempting to register with a bank with whom the end user does not maintain an account and where the bank only allows access to account holders. Next, the end user may have supplied improper or incorrect information. For example, a bank registration process might require a social security number, a password, a bank account number and the maiden name of the end user's mother; if the user entered an incorrect social security number, the registration process would fail. Finally, the PI provider may have altered the registration procedure for its Web site. In this situation, following the process supplied from the Provider store **320** would yield a failed registration. In the instance of any registration failure, the end user could be presented with the data initially supplied to the system for registration. The system could then ask the end user to double check the correctness of the information provided and to correct and resubmit the data if an error is found. A second failure resulting from the submission of identical requisite data might generate an error message presented to the end user

stating that either the end user is ineligible to access the selected PI from the selected PI provider or that alteration by the PI provider may have caused an error in registration. This second failure could also trigger a warning suggesting the need to potentially reconfigure the record for the PI provider in the Provider store **320**.

Ultimately, the user store **360** would contain a record for each end user. This record as previous described could be a database entry, one or more cookies or a file such as a PIC file. Each record would identify the selected PI providers along with the general access verification data needed and also under each PI provider would be a list of PI supplied and transactions supported by the particular PI provider of interest to the end user along with the additional data, if any, necessary to access that PI or execute that transaction. Specifically, duplicative information such as an end user's name would be centrally stored in the record once.

The end user configure component **330** also allows the end user to select one or more delivery destinations. One destination might be the end user's computer as exemplified by the client computer **220** running client software **270** in FIG. **2**; however, a computer is not the only destination contemplated by the present invention. The destination for PI delivery could include facsimile, electronic mail, telephone, conventional mail, pager, other wireless device such as a Palm Pilot (3 Com), Web page or channel, Web browser or other delivery mechanism. The present invention also contemplates indirect access of PI by the end user utilizing a Web site as an intermediary; however, such indirect access would not require the end user to specify a delivery destination unless additional delivery options were desired.

Further, access to the end user configure component **330** may occur through direct access to the PI engine via the Internet as contemplated by the client computer **220** running client software **270** in FIG. **2**; however, alternative methods of access are equally feasible. For example, the user might indirectly access the PI engine through the use of an intermediary Web site. A telephone interface to allow access to the end user configure component is another alternative.

With reference to FIG. **3**, the PI access/transact component **340** supports the update, acquisition and transaction functionality of the PI engine **240**. The PI access/transact component **340** is responsible for accessing and storing user PI and executing transactions authorized by the end user. When access or update is needed for a selected end user, the PI access/transact component **340** combines information from the Provider store **320** and the user store **360** to update end user PI in the PI store **280**. For each piece of PI requiring access or update, the PI access/transact component **340** looks up the access procedure and information needed for the particular PI in the Provider store **320**. The verification and access data is found in the user store **360**. The PI access/transact component **340** utilizes this information to connect to the PI provider's Web site across the Internet and to access the PI. Where multiple pieces of PI require updating or access, the accesses may occur in series or parallel.

Requested transactions would be similarly supported. For each transaction, the PI access/transact component **340** combines information from the Provider store **320** and the user store **360** to perform the requested transaction. The PI access/transact component **340** looks up the transaction procedure and information needed for the particular transaction in the Provider store **320**. The verification and access data is found in the user store **360**. The PI access/transact component **340** utilizes this information to perform the transaction across the Internet from the PI provider's Web site

A simulated Web client could perform access or transaction processes automatically supplying access and verification data as necessary. The manner in which such a simulated client access PI or execute transactions depends significantly upon the interaction method used on the PI provider Web site. If the Web site uses HTML forms and common gateway interface (CGI) applications, the PI access/transact component **340** can formulate a uniform resource locator (URL) to replicate the effect of actual form usage and submit this URL to the simulated Web client. The use of a URL to mimic an HTML form is equivalent to manually entering the data into the Web <FORM> element. See Kerven, Foust, Zakour, *HTML* 3.2 *Plus How-To*, Waite Group Press, 1997, pp. 559–569. If the Web site uses a mixture of HTML forms and Javascript functions, a simulated Web client with a modified Javascript interpreter could effectively access the PI or perform the transaction by following the PI access/transact process for the particular PI or transaction respectively. The access or transaction process to follow would be obtained from the record of the particular PI or transaction in the Provider store **320**. The Javascript interpreter in the simulated Web client would follow this procedure and supply the data found in the user store **360**. A similar process could be used if the PI provider Web site utilizes a Java applet. A Web client with a modified Java bytecode interpreter could effectively access PI or perform transactions by following process stored for the particular PI or transaction in the Provider store **320**. The bytecode interpreter would supply the data from the user store **360** rather than requiring interactive input from the end user. If the PI provider Web site utilizes a combination of forms, scripts and applets, the individual procedures above could be used in combination to accomplish the desired access.

In a preferred embodiment of such automated accesses or transactions, any necessary cookie data would be accepted and stored as needed by the PI access/transact component **340**. In many cases, cookie data is session specific and, therefore, of little long term utility. Cookies generated are used solely during these functions then discarded once the mining or transaction operation is complete.

In order to provide personal information to an end-user quickly after login, it is necessary for the PI access/transact component **340** to select an end user for data harvesting prior to the login of the end user. One approach to this solution is to update all of an end user's PI whenever the end user, directly or through an intermediary Web site, requests access to his/her PI. Another approach would be to update all of an end user's PI supplied by a particular provider whenever PI from that supplier is requested. Thus, the act of logging into the system by an end user effectively selects that end user for immediate PI update. However, this approach may result in the inefficient use of the PI Engine **240** resources.

Given the large number of potential users and providers, and the goal of providing the freshest data possible, another embodiment includes an algorithm developed to optimize the schedule in which end users are selected for data harvesting from a provider. This algorithm factors in the provider's update policy, the user's login habits, and the user-provider account characteristics. The proper application of the algorithm should ensure that PI is harvested as infrequently as possible for a given user, thus minimizing system resource consumption.

If the next provider update time and the next expected user login can be accurately predicted, a model can be

created that will allow for smarter harvesting. Rather than harvesting data for all users of a provider at once when the provider updates its site, the harvesting can be spread out over time based on expected login times of users and network activity profiles. For example, if Provider A updates its site on Friday night and a large number of users of that provider are not expected to login again until Monday morning, the harvesting load can be distributed across multiple days. This has the advantage of minimizing both the peak loading of the PI Engine **240** as well as consumption of the provider's bandwidth by the PI Engine **240**. To gain this optimization, the PI Engine **240** must maintain and refine models of each provider and user. Such data can be maintained in the provider store **310** and the user store **360** respectively.

Each time a user utilizes the PI Engine **240**, the time and date may be captured. Once a sufficient number of login times are accumulated, they may be analyzed with respect to day of month, day of week, and time of day. These are used in a model to predict the next expected user login. The model is then tested and refined with subsequent logins until a measurable degree of confidence is established. Once high confidence is determined, the user model is incorporated into the adaptive harvesting scheduler. Until a high confidence level is reached for a particular end user one of the aforementioned harvesting approaches may be used.

Each provider updates its site based on policy driven by their unique resources and business model. For any adaptive scheduler to work, the policy for each provider must be modeled. In some cases, the policy is self-evident. In others, it must be determined empirically. A provider's policy will most likely fall into one of the following categories:

Type I. Updated periodically for all users

Type II. Updated periodically relative to each user

Type III. Updated in a pseudo-random manner

The following three approaches may be used based upon provider type.

Type I Provider Policy Scheduling Algorithm

1. Assume users with a "no confidence" model have an immediate login time.
2. Order the users chronologically based on their predicted login time.
3. Shift the expected login time for all users back one hour.
4. Perform a density curve fit along temporal boundaries to get a polynomial function that can be used to determine the number of user accounts to harvest for a given epoch.
5. Perform an integral matching algorithm with the inverse of the network activity curve for the time period in question to adjust the distribution curve.
6. If possible, re-distribute peak harvesting time toward time zero to flatten the distribution curve.
7. Assign harvesting times to the sorted users according to the distribution curve.
8. Monitor time and harvest the user account when appropriate.

Type II Provider Policy Scheduling Algorithm

For each provider that falls into this category, an attribute of the user must be identified that determines when the personal information is updated. In some cases, the user may need to be queried for the information. In others, it can be determined from the harvested information. If the attribute cannot be established for a user via either of these means, the provider site may be monitored daily for changes in personal information until a pattern is established.

Since there is a natural, even distribution of accounts updated by a provider for a given day, a user's account can be harvested an hour before his expected login time. As in the Type I algorithm, users with a "no confidence" model should be immediately harvested.

Type III Provider Policy Scheduling Algorithm

This type of policy is the most difficult of all. Since the provider updates a user's account in a non-deterministic manner, a decision must be made for each provider as to the criticality of the information relative to the user. For those highly critical providers, each user account should be harvested daily, perhaps even more frequently. For those less critical providers, user accounts should be harvested less frequently and possible when overall system activity is low.

The PI deliver component **350** is responsible for formatting and delivering the PI to the end user. Usually delivery will only occur subsequent to updating all stale PI. The PI will be delivered to one or more destinations (e.g. facsimile, telephone, pager, Web browser, e-mail, etc.) as specified in the user store **360** except where the PI is accessed via an intermediary Web site. Where the destination is not an intermediary Web site, the PI deliver component **350** performs all formatting necessary to deliver the PI to the appropriate destinations. For example, where the destination is a Web browser, the PI would be formatted as an HTML document, or where the destination is a telephone, the PI would be submitted for voice synthesis and transmission.

In the case of an intermediary Web site, the PI is delivered in a format configurable by the intermediary Web site. FIG. **5** pictorial illustrates a possible embodiment of the current invention utilizing an intermediary Web site. An end user **210** utilizes a client computer **220** to access an intermediary Web site **510** across the Internet **230**. The end user **210** logs into the intermediary Web site **510**. The intermediary Web site **510** contacts the PI engine **240** across the Internet **230** and directly receives the end user's PI updated as required from the PI provider Web sites **250**. The intermediary Web site **510** receives the PI, incorporates it into pages according to its particular formatting style and graphical user interface and delivers these pages to the end user **210**. The use of the PI engine **240** is transparent to the end user **210**. Further, an intermediary Web site **510** serving aggregate PI to an end user **210** may, and most likely will, simultaneously serve as a PI provider.

In another embodiment, this formatting occurs via a dynamic HTML generation system combining stylistic and layout information from a variety of sources. The PI deliver component **350** generates custom HTML pages dynamically. These pages are customized based on a number of stylistic factors (such as background color, foreground color, font size, color and style, page layout, etc) from a variety of sources and content from a variety of sources. Information providers, distributors, the end user, the PI deliver component **350** or any combination of these sources, or other relevant sources, may provide customization factors used in the page generation. Finally, each HTML page must be filled in with data. The data used in such pages may originate from such sources as information providers, distributors, the end user, the PI deliver component **350** or any combination of these sources, or other relevant sources. The required solution is a system representing a generic algorithm for performing such HTML generation at run-time. The style and content may be provided in any suitable format such as the Extensible Stylesheet Language (XSL), as specified by W3C in http://www.w3.org/TR/WD-xsl/, which is expressly incorporated herein by reference in its entirety, and/or the Extensible Markup Language (XML) as specified by W3C

in http://www.w3.org/TR/REC-xml, which is expressly incorporated herein by reference in its entirety, or other suitable formatting standard. The key requirements for such a system are complete encapsulation of the problem domain and run-time efficiency.

In preferred embodiments, the solution is based on the following basic model as depicted in FIG. **8**:

1. Six sets of customization factors are identified: distributor content **810**, provider content **820**, distributor style specification **830**, provider style specification **840**, user-specific content **850** and user-specific style **860**.
2. Each set of customization factors **810–860** is considered a separate, independent and required input to the run-time system **870** that performs dynamic page generation.
3. Each input **810–860** will be in form of an XML stream.
4. Output **880** will be in form of an HTML stream.
5. The dynamic page generation system **870** will produce valid output **880** for each set of six valid inputs **810–860**.

FIG. **9** illustrates an actual run-time sequence of input processing by such a system **870**:

1. Distributor content **810** is combined with provider content **820** and with user-specific content **850** to produce a complete content specification **930** by the content merger unit **910**.
2. Distributor style **830** is combined with provider style **840** and with user-specific style **860** to produce a complete style specification **940** by the style merger unit **920**.
3. The style specification **940** is applied by the style applicator **950** to content specification **930** in order to produce the resulting page **880**.

In order to completely encapsulate the problem domain, the following requirements must be placed on the system **870**:

1. Each XML input **810–860** is a valid XML stream.
2. All content specifications **810, 820** and **850** are valid with respect to the same Document Type Definition.
3. All style specifications **830, 840** and **860** are valid with respect to the same Document Type Definition (such as the XSL DTD standard).
4. The merging units **910** and **920** whose task is to take two or more XML streams and produce a combined XML output must be able to produce such output for any set of valid XML inputs.

Another method of performing this task would be to format PI as HTML elements with predefined CLASS attributes. The intermediary Web site receiving these elements could dynamically include them in page forwarded to the end user of the PI. The pages incorporating such elements could include different style information associated with the predefined CLASS set. Level 1 cascading style sheet convention could be used to implement such configurability. See Kerven, Foust, Zakour, *HTML* 3.2 *Plus How-To*, Waite Group Press, 1997, pp. 651–693; Walsh, "An Introduction to Cascading Style Sheets," *World Wide Web Journal*, Winter 1997, pp. 147–156. This option requires minimal programmatic support by the intermediary Web site but restricts to some degree the intermediary Web sites flexibility in presenting the PI to the end user.

Alternatively, an intermediary Web site could develop an application utilizing a standardized application programming interface (API) to directly access the PI data. In this instance, the PI deliver component **350** could either be bypassed or potentially used as the component responsible for servicing API requests for data. Under this model, the intermediary Web site would be responsible for all formatting decisions with respect to the raw PI data. This implementation option requires additional programmatic support by the intermediary Web site but allows for greater flexibility in the use of the raw PI.

The ability to utilize an intermediate Web site to deliver PI is of significant utility. This capability allows an end user already familiar with an existing PI provider to access not only the PI associated with the particular PI provider but also all PI from other PI providers in the comfort of a familiar user interface, namely the existing PI provider Web site. In this situation, the request for PI would directly originate with the intermediary PI provider Web site and indirectly from the end user. Security measures would restrict access to authorized intermediate Web site access. These measure might include verification of the end user and the intermediate Web site. Further, verification of the association between the end user and the particular intermediate Web site might also be required for additional security.

In addition, the use of an intermediary Web site also supports a novel transaction model. In this transaction model, the intermediary site subsidizes, or fully compensates, the PI engine administrator for services provided to the end user. These transactions are facilitated via the auditing and tracking capabilities of the PI engine. These capabilities allow the calculation of per user fees, per transaction fees, per access fees or some combination thereof to be assessed. The assessed values could be directly charged to the intermediary Web site. Alternatively, such values could be debited from a minimum monthly fee charged to the intermediary Web site with any fees beyond the minimum charged directly to the intermediary Web site.

FIG. **11** depicts a flowchart of a typical process according to the described model. The intermediary Web site pays a minimum monthly fee in step **1110**. In step **1120**, the PI engine audits and tracks end user usage via the intermediary Web site. The audited usage is used to assess a fee on a per user, per access, per transaction or combination basis. In step **1130**, this audited amount is debited from the fee paid in step **1110**. In step **1140**, the intermediary Web site is charged for any fees in excess of the minimum fee paid.

Often an end user may require access to the underlying Web page generated by the provider of a particular piece of PI. The delivery component may deliver not only the PI but also an access point directly to the provider's page supplying that PI. The access point may take the form of a link, a form button or some other interactive access mechanism.

Such an access point significantly improves the efficiency of accessing the underlying page by the end user as exhibited by FIG. **7**. In the traditional process **100** for accessing PI, the end user must proceed through numerous intermediary pages requiring a variety of often tedious interactions before reaching the desired page.

The end user must first identify the Provider **110**. Next, the end user must locate the Provider's Web address **120**. Then, the user the requests the Provider's login page **130**. If the end user does not remember the requisite information, this information must be found, or the desired information will remain inaccessible via the Web. The end user then navigates the Provider's Web site **140**. This often entails visiting the Provider's main page **710** followed by viewing a variety of intermediate pages on the Provider's site **720**. The end user may have to backtrack several times to the main page **710** or accidentally leave the system entirely forcing a second login **140** before finally locating the desired information **150**.

Utilizing springboard technology, the entire process **750** is streamlined into the single click of an access point. The delivery component of the PI Engine delivers an access point to the Provider's underlying page along with the PI. As a consequence, the end user need only perform a single interaction with the PI presentation page **760**. This interaction immediately performs the requisite interactions with the Provider's Web site to bring the user to the desired underlying Web page **150**.

In one embodiment, this springboard technology could be implemented utilizing a Java applet. With respect to FIG. **2**, the applet would be downloaded from the PI Host **290** by the end user's client software **270**, usually a Web browser, and executed locally by the end user's computer **220**. The applet would drive the client software **270** to the desired page. Such an applet could retrieve procedures and data for driving the client software from the Provider store **310** and the User store **360**.

In a further embodiment, the PI engine **240** could act as a proxy server directly accessing the Provider store **310** and the User store **360** as required. When the PI engine **240** receives the request to jump to the source of a particular piece of PI, the engine performs the necessary actions to navigate to the desire page and forwards the desired page to the end user's computer **220**. Further interactions with the page might require additional proxying by the PI engine **240** as accumulated cookie data may reside on the PI Host **290**. This embodiment is limited to use in handling standard HTTP traffic rather than secure HTTP traffic.

In a preferred embodiment, the springboard provides the end user with automated login into the PI Provider site **250** and allows the end user **210** to navigate via the client software **270**. This automated login could be accomplished through the utilization of a hypertext transfer protocol (HTTP) redirect. Upon receiving the a springboard access request from the end user **210** via the client software **270**, the PI Host **290** requests the login page from the PI Provider site **250** targeted by the springboard access. The PI engine **240** running on the PI Host **290** receives this login page and constructs a login request by accessing the proper data in the Provider store **310** and the User store **360**. The login request is embedded in the HTTP redirect which is forward to the client software **270**. The client software **270** is redirected to the targeted PI Provider site **250**, and the end user **210** is automatically logged into this site.

Alternatively, this functionality could be implemented via a Java applet as described above. In addition, the PI engine **240** could generate a Javascript page containing the pertinent login request rather than an HTTP redirect. The Javascript page could be returned to the client software **270**. This page would then be executed by the client software **270** to accomplish the automated login.

The PI engine **240** of FIG. **3** may also include a site monitor **370** processing component. This component would systematically monitor supported PI provider Web sites for changes. This component enhances the ability of the system to identify alterations in PI provider Web site procedures, data requirements and cookies requirements. This component increases system efficiency by supplementing or supplanting alteration identification via feedback from the PI access/transact component **340**.

A further embodiment of the present invention might support the localize manipulation of PI. This could be accomplished where the client software **270** running on the client computer **220** in FIG. **2** is a specialized Web client rather than a general Web client such as Netscape. This specialized client might utilize Web channel technology to automate the local PI download and update processes. Where the PI store is implemented via the aforementioned cookie architecture, this specialized client may provide direct local access to stored PI.

In another embodiment, the PI engine **240** of FIG. **3** might support both system supported PI providers as well as PI providers specific to particular end users. In this embodiment, an end user is not limited to PI available from PI providers present in the Provider store **310**. For an end user to add PI provided by a non-supported PI provider, the end user would access the Baseline configure component **320** and create a configuration for the non-supported PI provider. The PI provider and PI configuration along with the verification and access data would be stored along with the user's record in the user store **360**.

A further embodiment of the present invention supports the inclusion of PI transaction procedures and access requirements in the Provider store **310** of FIG. **3**. The end user specific information necessary to realize such a transaction would reside with the user record in the user store **360**. The functionality of the PI access/transact component **340** would expand to support the performance of transactions. This additional functionality could be supported in a manner similar to the procedure described above with respect to performance of access utilizing a simulated Web client. A further feature of this embodiment would include automated or semi-automated account management by providing trigger events to automatically initiate a transaction.

For instance, with reference to FIG. **2** an end user **210** would be able to maintain his/her accounts online through the PI Engine **240**. If an information provider has the capability of receiving payments online, the PI Engine **240** could support complete or partial automation of such transactions. If there is a billing due date for a certain information provider, PI Engine **240** could flag that information and send email to the end user **210** notifying him/her of the bill due. Thus, the user will not have to check each of his/her providers individually for due date information. The PI Engine **240** could also automated payments on a limited range of billing amount for providers who allow payments over their Web servers **260**, then send an email to the user with the notification of payment.

Due date acquisition could be accomplished utilizing the PI access/transact component **340** seen in FIG. **3**. The due date information would be available to the end user via any delivery means supported by the PI deliver component **350**. The PI access/transact component **340** would use standard e-commerce bill-paying methods to pay the user's bill/s to the provider if he/she chooses. Once the bill is paid, then an email notification will be sent to the user with the provider information and payment information. The user can specify the range of amount stored in the user store **360** that will be paid automatically. If the bill exceeds the amount specified by the user, then PI engine will simply send out an email notification to the user instead of paying the bill automatically.

The embodiments described above are given as illustrative examples only. It will be readily appreciated that many deviations may be made from the specific embodiment disclosed in this specification without departing from the invention. Accordingly, the scope of the invention is to be determined by the claims below rather than being limited to the specifically described embodiments above.

What is claimed is:

1. A method for automated access to personal information associated with an end user, wherein the personal information is stored on a personal information provider, the method comprising the steps of:

(a) presenting on a client computer associated with the end user and in communication with the personal information provider via a network a representation of personal information and a link corresponding to the personal information stored on the personal information provider; and

(b) upon activation of the presented link, transmitting a page containing a form which includes login information that upon opening by the client computer redirects the client computer to a post login page on the personal information provider in accordance with a protocol, the personal information accessible to the client computer in accordance with the protocol also being accessible by the end user via the network independently of the method for automated access to personal information.

2. A method for automated access to personal information associated with an end user, wherein the personal information is stored on a personal information provider, the method comprising the steps of:

(a) presenting on a client computer associated with the end user and in communication with the personal information provider via a network a representation of personal information and a link corresponding to the personal information stored on the personal information provider;

(b) upon activation of the presented link, downloading an application to the client computer, wherein the downloaded application upon execution on the client computer performs the steps of:

(i) connecting to the personal information provider;

(ii) navigating to the personal information on the personal information provider using a protocol for instructing the client computer how to access the personal information via the network, the personal information accessible to the client computer using the protocol also being accessible by the end user via the network independently of the method for automated access to personal information; and

(iii) presenting the personal information to the user of the client computer.

3. The method of claim **2**, and further comprising the steps of:

(c) transmitting end user data associated with the end user to the client computer; and

(d) transmitting personal information provider data associated with the personal information provider to the client computer.

4. The method of claim **3**, wherein the step of transmitting personal information provider data associated with the personal information provider comprises transmitting a navigation script corresponding to the personal information.

5. The method of claim **4**, wherein the step of transmitting end user data associated with the end user comprises transmitting end user data associated with the end user based on the transmitted navigation script.

6. The method of claim **2**, and further comprising the step of generating an application for downloading to the client computer based on personal information provider data associated with the personal information provider, on the personal information and on end user data associated with the end user.

7. A method for automated access to personal information associated with an end user, wherein the personal information is stored on a personal information provider, the method comprising the steps of:

(a) presenting on a client computer associated with the end user and in communication with the personal information provider via a network a representation of personal information and a link corresponding to the personal information stored on the personal information provider;

(b) upon activation of the presented link, driving the client computer to the personal information stored on the personal information provider by performing the steps of:

(i) connecting to the personal information provider;

(ii) navigating to the personal information on the personal information provider using a protocol for instructing the client computer how to access the personal information via the network, the personal information accessible to the client computer using the protocol also being accessible by the end user via the network independently of the method for automated access to personal information;

(iii) presenting the personal information to the user of the client computer; and

(iv) proxying subsequent requests from the client computer to the personal information provider.

8. A computer-readable storage device storing instructions that upon execution cause a processor to automatically access personal information associated with an end user, wherein the personal information is stored on a personal information provider by performing the steps comprising of:

(a) presenting on a client computer associated with the end user and in communication with the personal information provider via a network a representation of personal information and a link corresponding to the personal information stored on the personal information provider;

(b) upon activation of the presented link, downloading an application to the client computer, wherein the downloaded application upon execution on the client computer performs the steps of:

(i) connecting to the personal information provider;

(ii) navigating to the personal information on the personal information provider using a protocol for instructing the client computer how to access the personal information via the network, the personal information accessible to the client computer using the protocol also being accessible by the end user via the network independently of automatic access by the processor of personal information caused by execution of the instructions; and

(iii) presenting the personal information to the user of the client computer.

9. A system for automated access to personal information associated with an end user, wherein the personal information is stored on a personal information provider in communication with a client computer via a network, the system comprising:

(a) a user store for storing data associated with the end user, the data associated with the end user including information identifying a plurality of information providers storing personal information associated with the end user;

(b) a personal information provider store for storing data associated with the personal information provider, the data associated with the personal information provider including a protocol for instructing a processor how to access the personal information via the network; and

(c) a processor in communication with the user store and the personal information provider store, the processor for performing the steps of:

(i) presenting on a client computer associated with the end user a representation of personal information and

a link corresponding to the personal information stored on the personal information provider;

(ii) upon activation of the presented link, downloading an application to the client computer, wherein the downloaded application upon execution on the client computer performs the steps of:

(A) connecting to the personal information provider;

(B) navigating to the personal information on the personal information provider using the protocol, the personal information accessible to the client computer using the protocol also being accessible by the end user via the network independently of the system for automated access to personal information; and

(C) presenting the personal information to the user of the client computers.

* * * * *

US006560640B2

(12) **United States Patent**
Smethers

(10) **Patent No.: US 6,560,640 B2**
(45) **Date of Patent: *May 6, 2003**

(54) **REMOTE BOOKMARKING FOR WIRELESS CLIENT DEVICES**

(75) Inventor: **Paul A. Smethers**, Cupertino, CA (US)

(73) Assignee: **Openwave Systems, Inc.**, Redwood City, CA (US)

(*) Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/235,231**

(22) Filed: **Jan. 22, 1999**

(65) **Prior Publication Data**

US 2003/0055870 A1 Mar. 20, 2003

(51) **Int. Cl.**[7] **C06F 17/30**
(52) **U.S. Cl.** **709/219**; 709/228
(58) **Field of Search** 358/473; 707/10, 707/104; 709/203, 206, 218, 219, 236, 228

(56) **References Cited**

U.S. PATENT DOCUMENTS

4,617,657	A		10/1986	Drynan et al.
5,199,104	A		3/1993	Hirayama
5,278,673	A	*	1/1994	Scapa et al. 358/473
5,425,077	A		6/1995	Tsoi
5,481,539	A		1/1996	Hershey et al.
5,692,032	A		11/1997	Seppänen et al.
5,737,560	A		4/1998	Yohanan
5,751,708	A		5/1998	Eng et al.
5,761,280	A		6/1998	Noonen et al.
5,797,098	A		8/1998	Schroeder et al.
5,802,516	A		9/1998	Shwarts et al.
5,809,415	A		9/1998	Rossmann
5,895,471	A	*	4/1999	King et al. 707/104
5,930,472	A	*	7/1999	Smith 709/203
6,049,831	A	*	4/2000	Gardell et al. 709/236
6,138,151	A	*	10/2000	Reber et al. 709/219
6,138,158	A	*	10/2000	Boyle et al. 709/225
6,173,316	B1	*	1/2001	De Boor et al. 709/218
6,182,113	B1	*	1/2001	Narayanaswami 709/203
6,208,839	B1	*	3/2001	Davani 455/31.3
6,243,739	B1		6/2001	Schwartz et al.
6,272,129	B1	*	8/2001	Dynarski et al. 370/356
6,321,257	B1	*	11/2001	Kotola et al. 709/219

FOREIGN PATENT DOCUMENTS

JP	10083241	3/1998
WO	97 22212	6/1997
WO	98 11744	3/1998

OTHER PUBLICATIONS

HDML 2.0 Language Reference, Version 2.0, Unwired Planet, Inc., Software Developer Kit, Jul. 1997.
"HDTP Specification", Version 1.1–Draft, Unwired Planet, Inc., Jul. 15, 1997.
UP.Browser™ User Handbook, Unwired Planet, Inc., Nov. 1977.
"Wireless Application Protocol Architecture Specification" (WAP Architecture), Version 30, Apr. 1998.

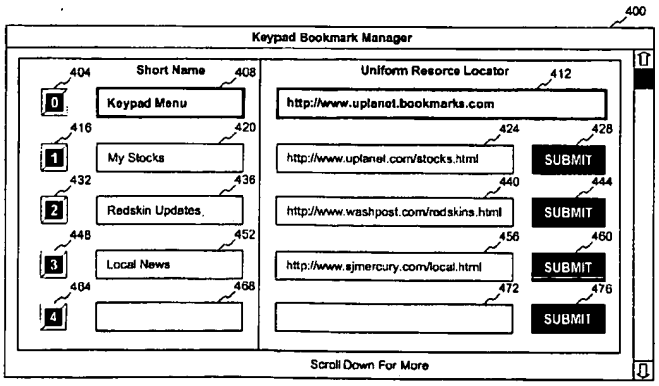
* cited by examiner

Primary Examiner—Christine Oda
Assistant Examiner—Walter Benson
(74) *Attorney, Agent, or Firm*—Beyer Weaver & Thomas, LLP

(57) **ABSTRACT**

Improved techniques that enable wireless devices to implement bookmarks with improved transmission efficiency, reduced user navigation and/or reduced amounts of memory resources are disclosed. One aspect of the improved techniques pertains to use of a compact request from a wireless device to an intermediate server when requesting a document or file by selection of a bookmark. Another aspect of the improved techniques is the ability of a user to select a bookmark to request the associated document or file with reduced user interaction (e.g., a single button action). Still another aspect of the improved techniques is that memory resources of the wireless devices need not be consumed to store network addresses (e.g., URLs) for the bookmarks.

34 Claims, 9 Drawing Sheets



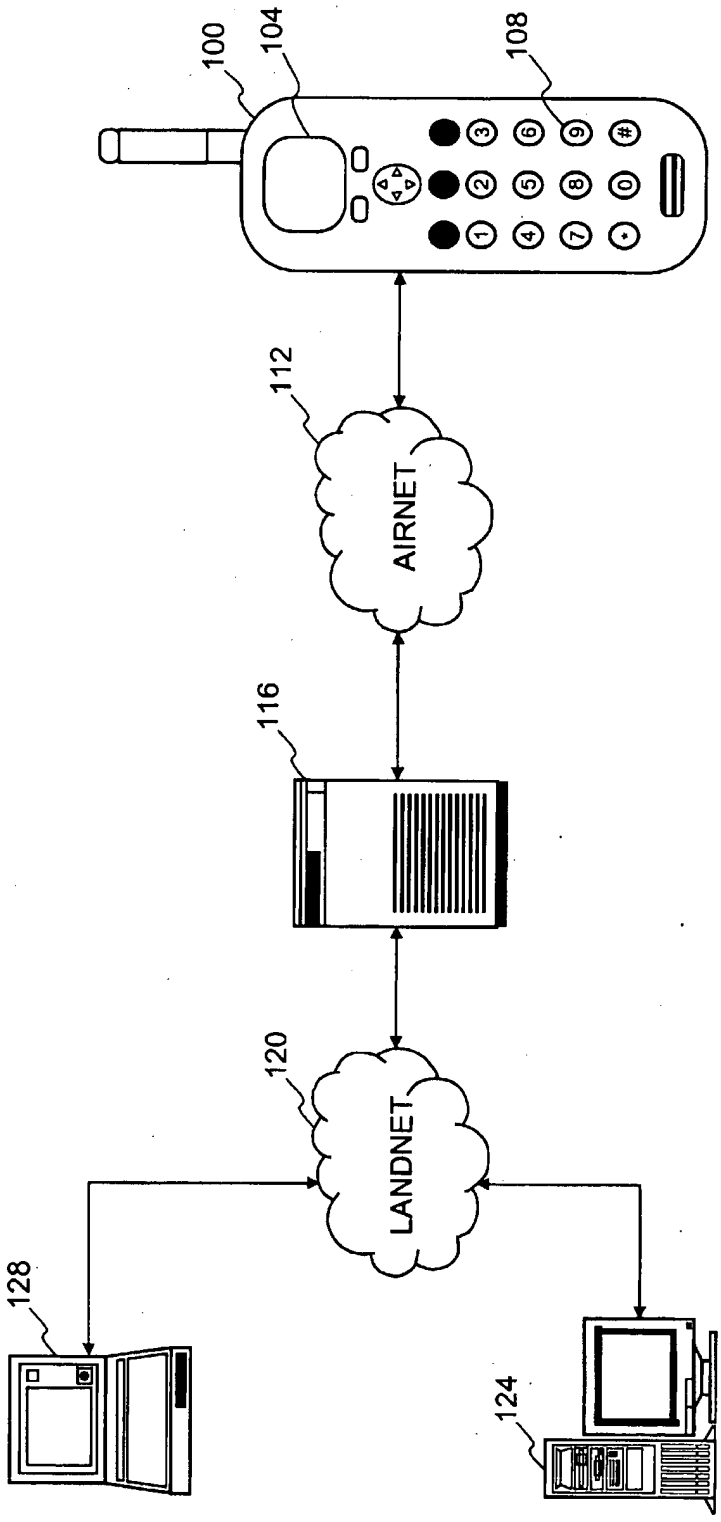


Fig. 1

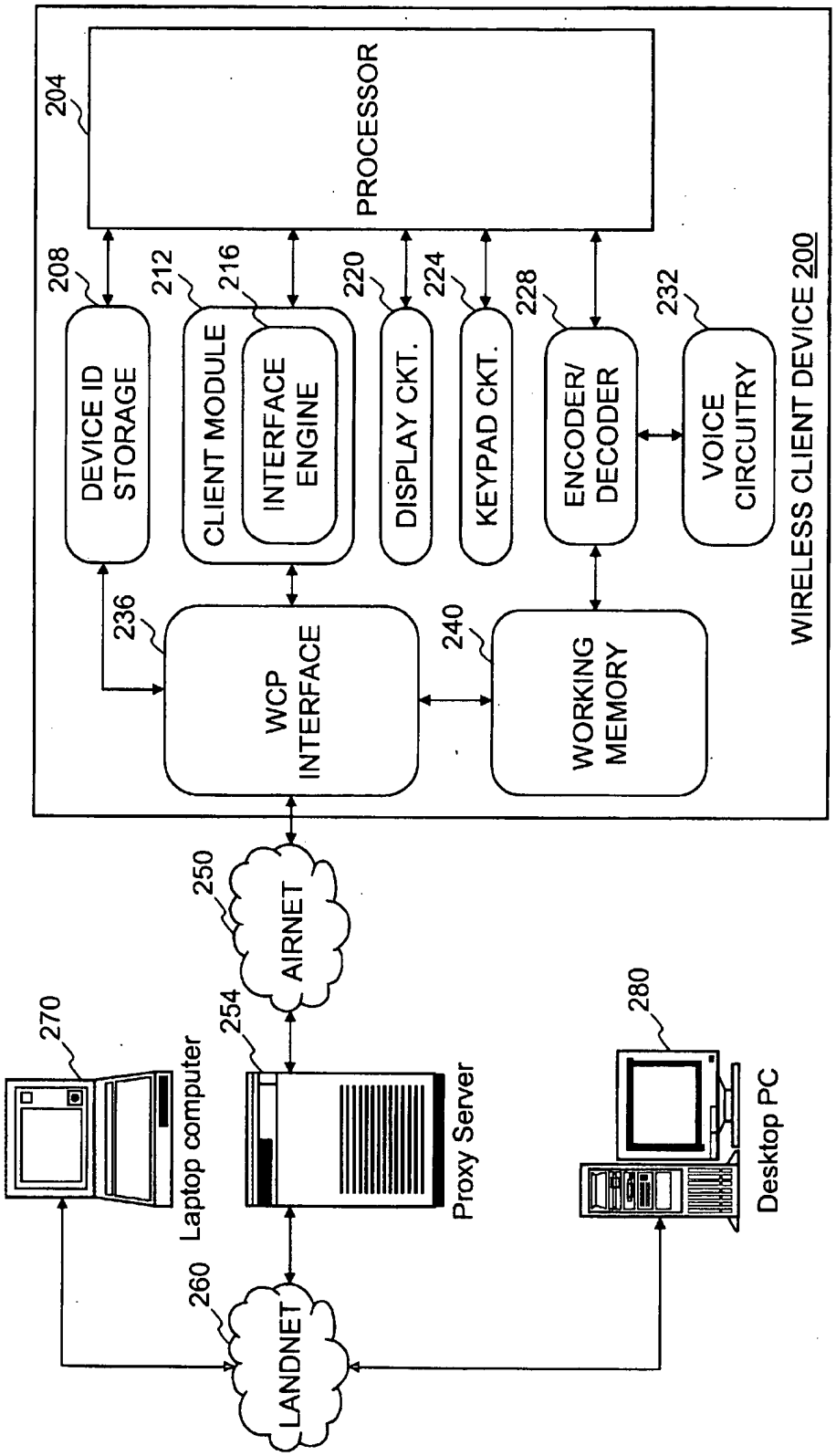

204
PROCESSOR
208
DEVICE ID STORAGE
212
CLIENT MODULE
216
INTERFACE ENGINE
220
DISPLAY CKT.
224
KEYPAD CKT.
228
ENCODER/ DECODER
232
VOICE CIRCUITRY
236
WCP INTERFACE
240
WORKING MEMORY
WIRELESS CLIENT DEVICE 200
250
AIRNET
254
Proxy Server
270
Laptop computer
280
Desktop PC
260
LANDNET

Fig. 2

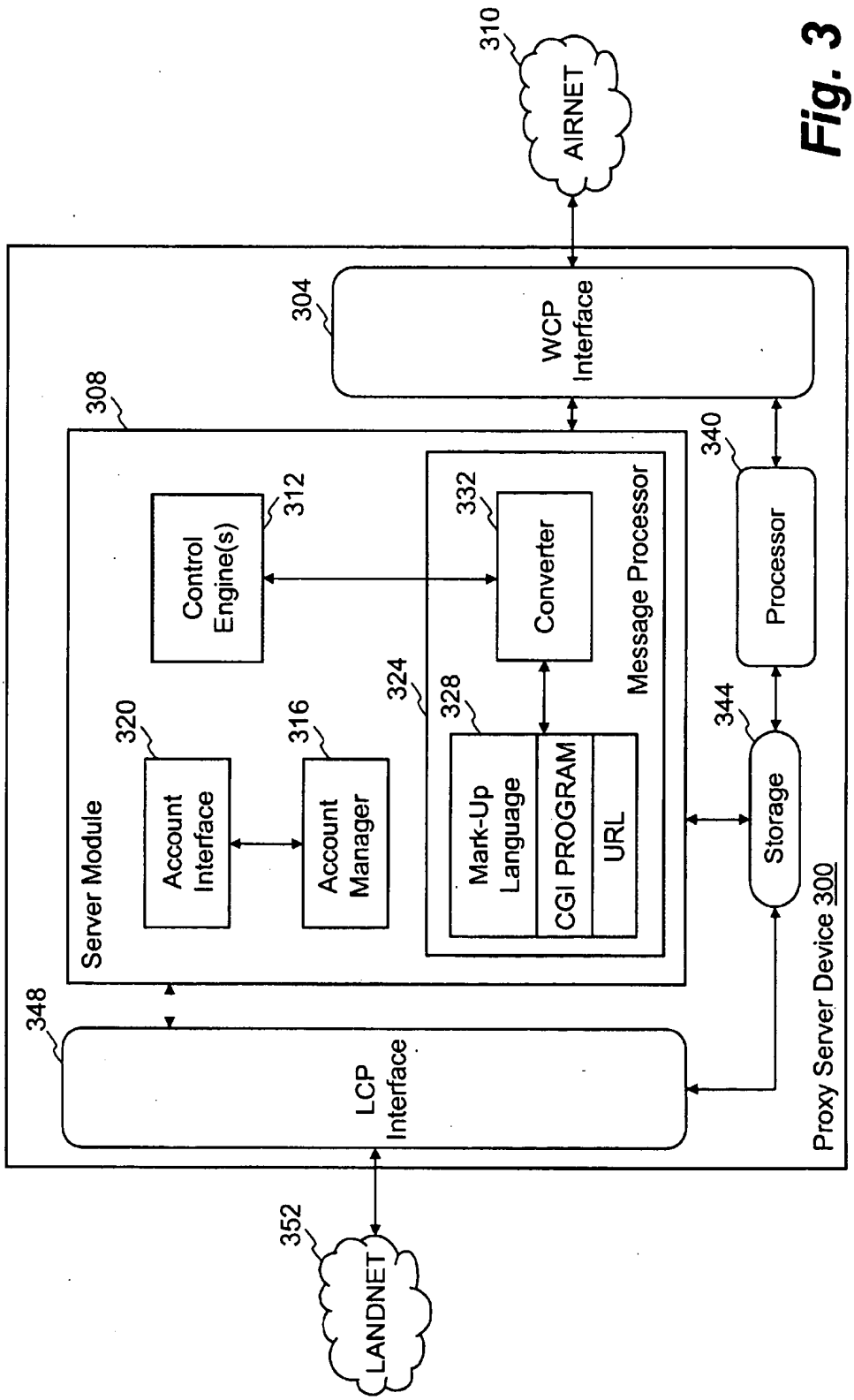

310
AIRNET
304
WCP Interface
308
340
Processor
312
Control Engine(s)
332
Converter
Message Processor
324
328
Mark-Up Language
CGI PROGRAM
URL
320
Account Interface
316
Account Manager
344
Storage
Server Module
348
LCP Interface
Proxy Server Device 300
352
LANDNET

Fig. 3

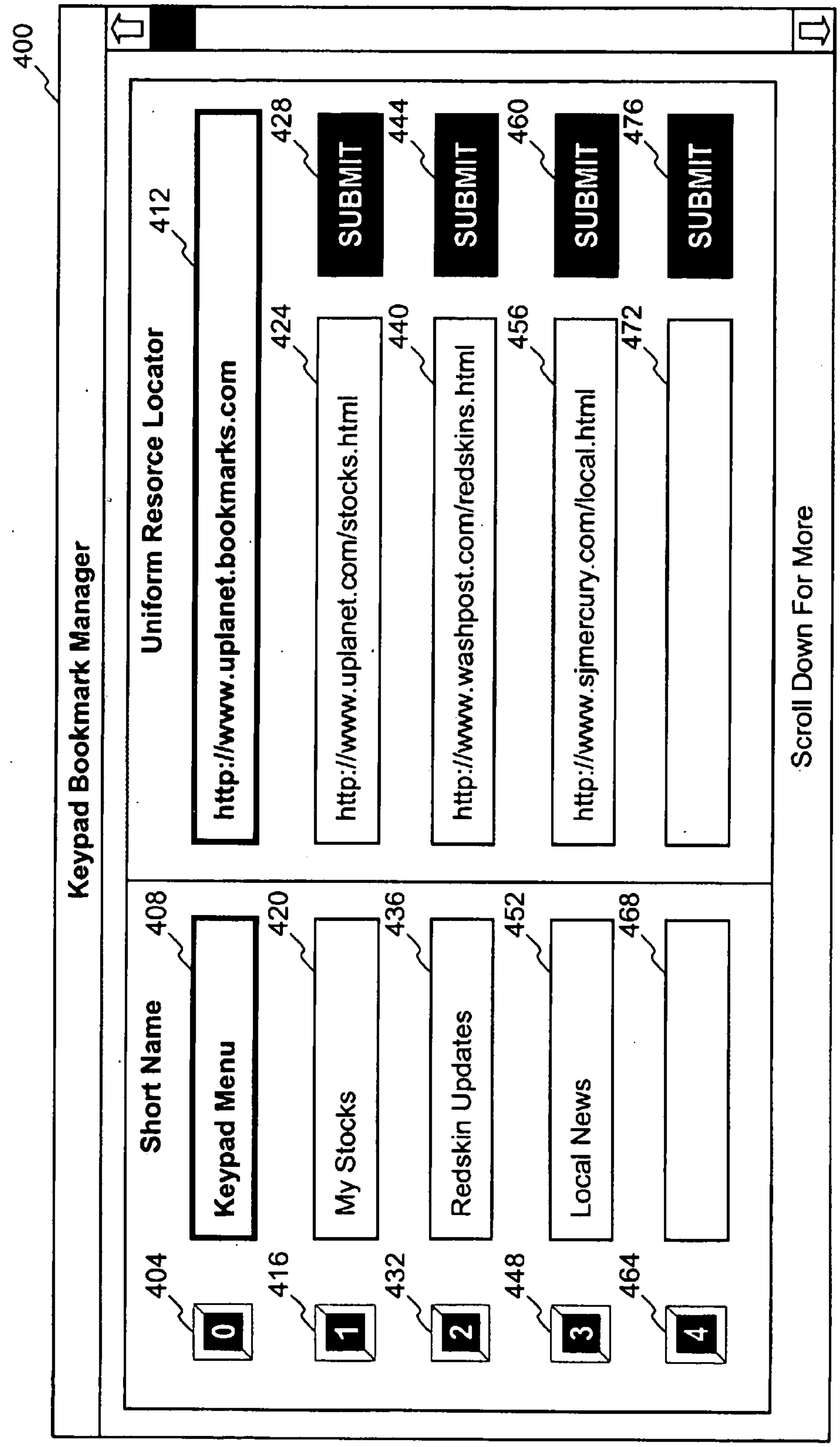

400
Keypad Bookmark Manager
Short Name
Uniform Resorce Locator
404
0
408
Keypad Menu
412
http://www.uplanet.bookmarks.com
416
1
420
My Stocks
424
http://www.uplanet.com/stocks.html
428
SUBMIT
432
2
436
Redskin Updates
440
http://www.washpost.com/redskins.html
444
SUBMIT
448
3
452
Local News
456
http://www.sjmercury.com/local.html
460
SUBMIT
464
4
468
472
476
SUBMIT
Scroll Down For More

Fig. 4

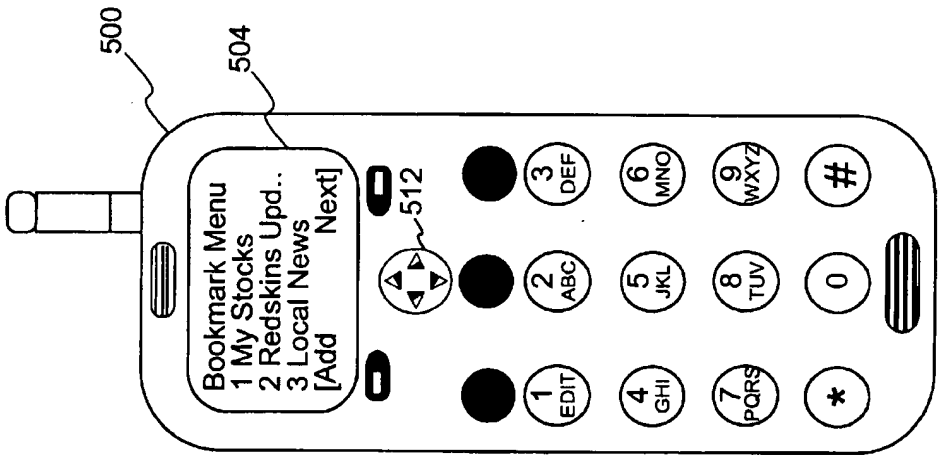

500
504
512
Bookmark Menu
1 My Stocks
2 Redskins Upd..
3 Local News
[Add Next]
1 EDIT
2 ABC
3 DEF
4 GHI
5 JKL
6 MNO
7 PQRS
8 TUV
9 WXYZ
0
*
#

Fig. 5

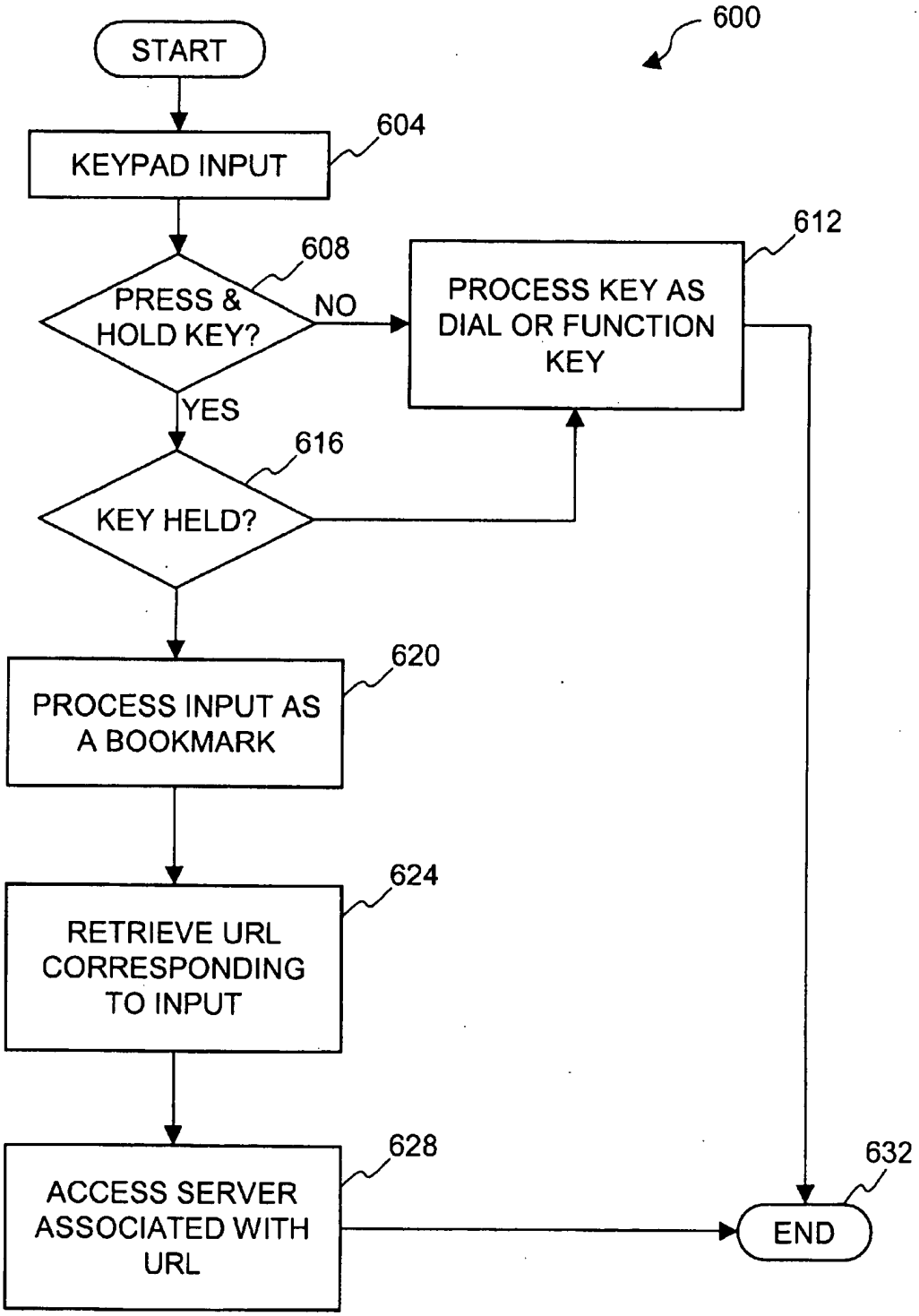

600
START
KEYPAD INPUT
604
PRESS & HOLD KEY?
608
NO
PROCESS KEY AS DIAL OR FUNCTION KEY
612
YES
KEY HELD?
616
PROCESS INPUT AS A BOOKMARK
620
RETRIEVE URL CORRESPONDING TO INPUT
624
ACCESS SERVER ASSOCIATED WITH URL
628
END
632

Fig. 6

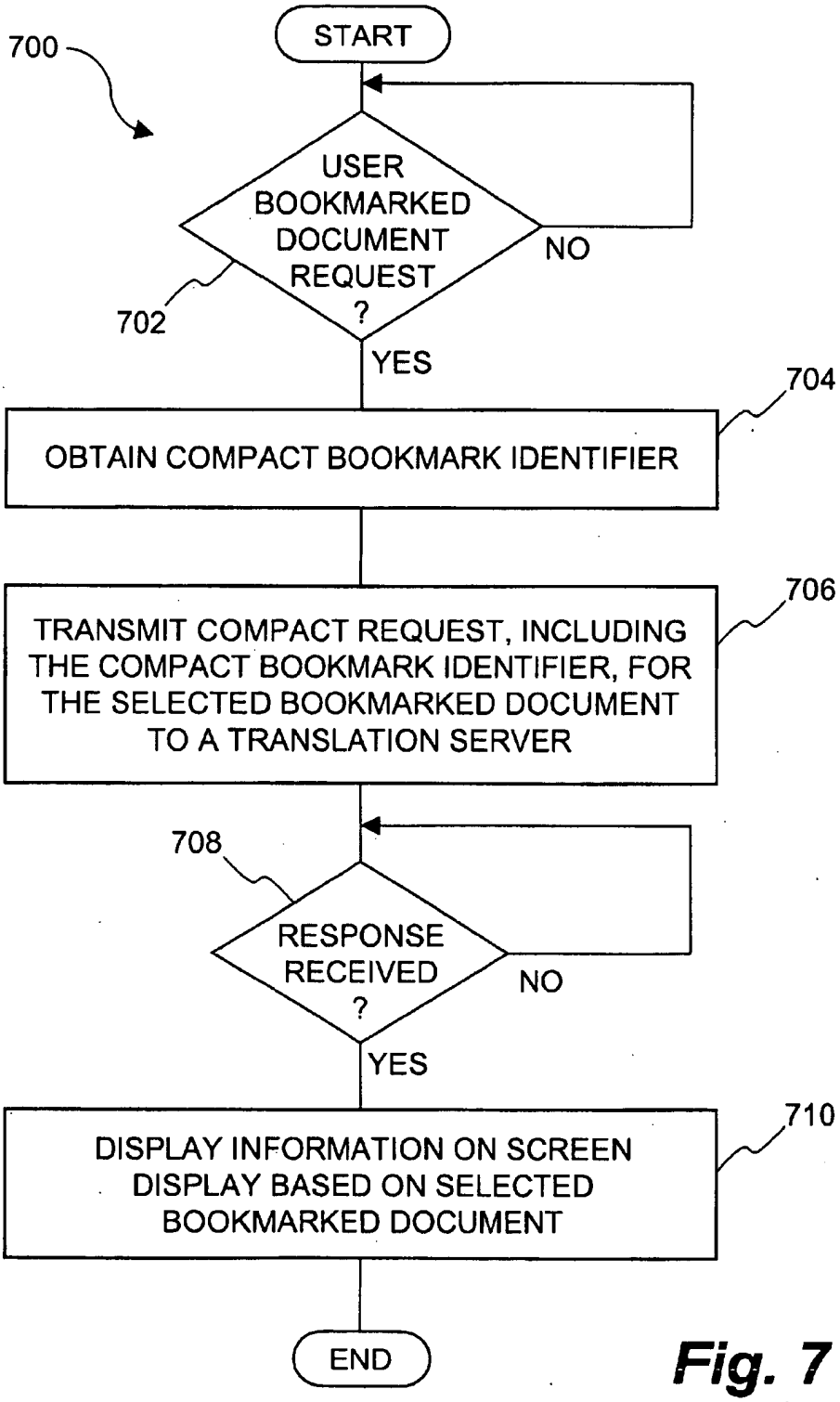

START
700
USER BOOKMARKED DOCUMENT REQUEST ?
NO
702
YES
704
OBTAIN COMPACT BOOKMARK IDENTIFIER
706
TRANSMIT COMPACT REQUEST, INCLUDING THE COMPACT BOOKMARK IDENTIFIER, FOR THE SELECTED BOOKMARKED DOCUMENT TO A TRANSLATION SERVER
708
RESPONSE RECEIVED ?
NO
YES
710
DISPLAY INFORMATION ON SCREEN DISPLAY BASED ON SELECTED BOOKMARKED DOCUMENT
END

Fig. 7

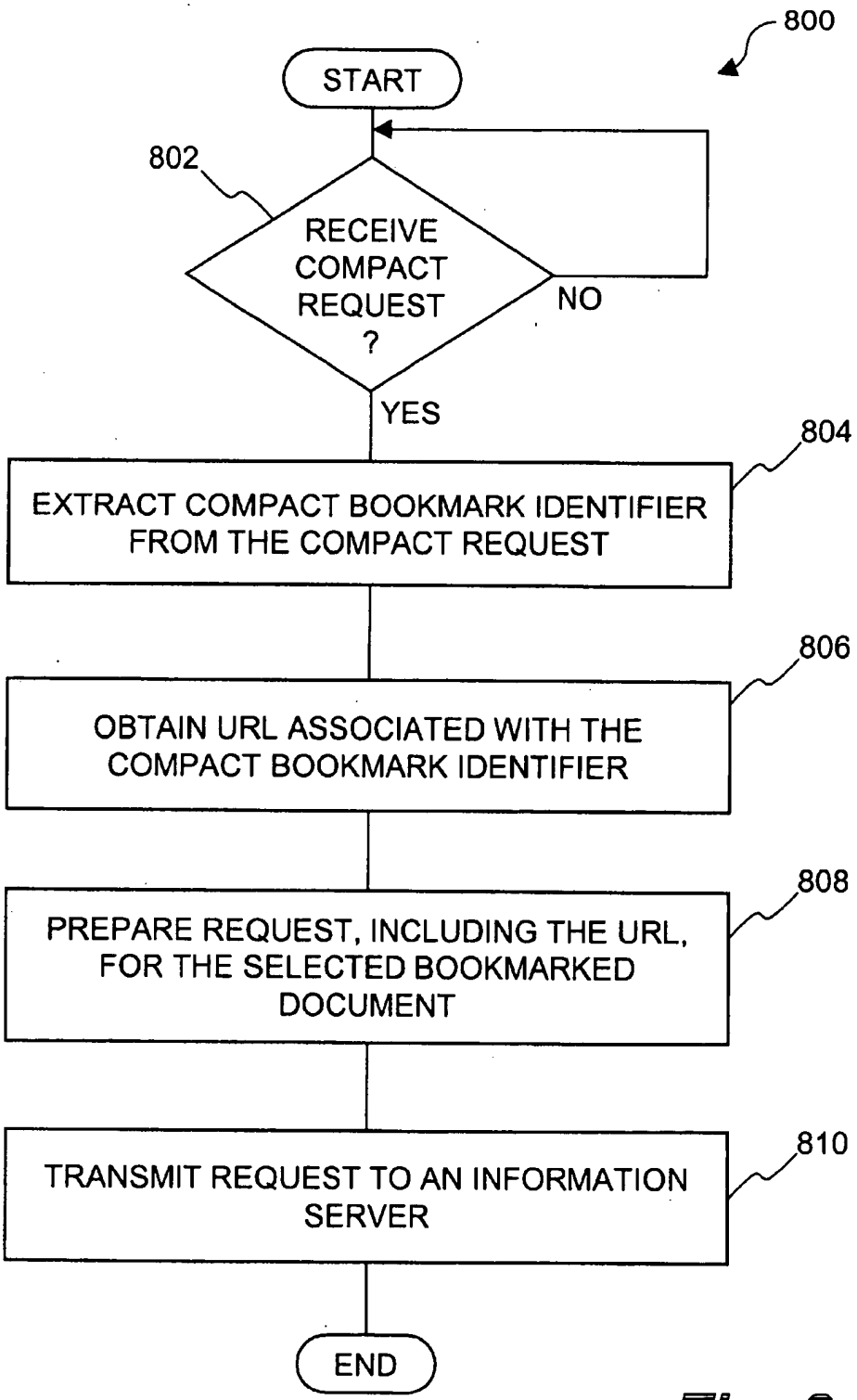

800
START
802
RECEIVE COMPACT REQUEST ?
NO
YES
804
EXTRACT COMPACT BOOKMARK IDENTIFIER FROM THE COMPACT REQUEST
806
OBTAIN URL ASSOCIATED WITH THE COMPACT BOOKMARK IDENTIFIER
808
PREPARE REQUEST, INCLUDING THE URL, FOR THE SELECTED BOOKMARKED DOCUMENT
810
TRANSMIT REQUEST TO AN INFORMATION SERVER
END

Fig. 8

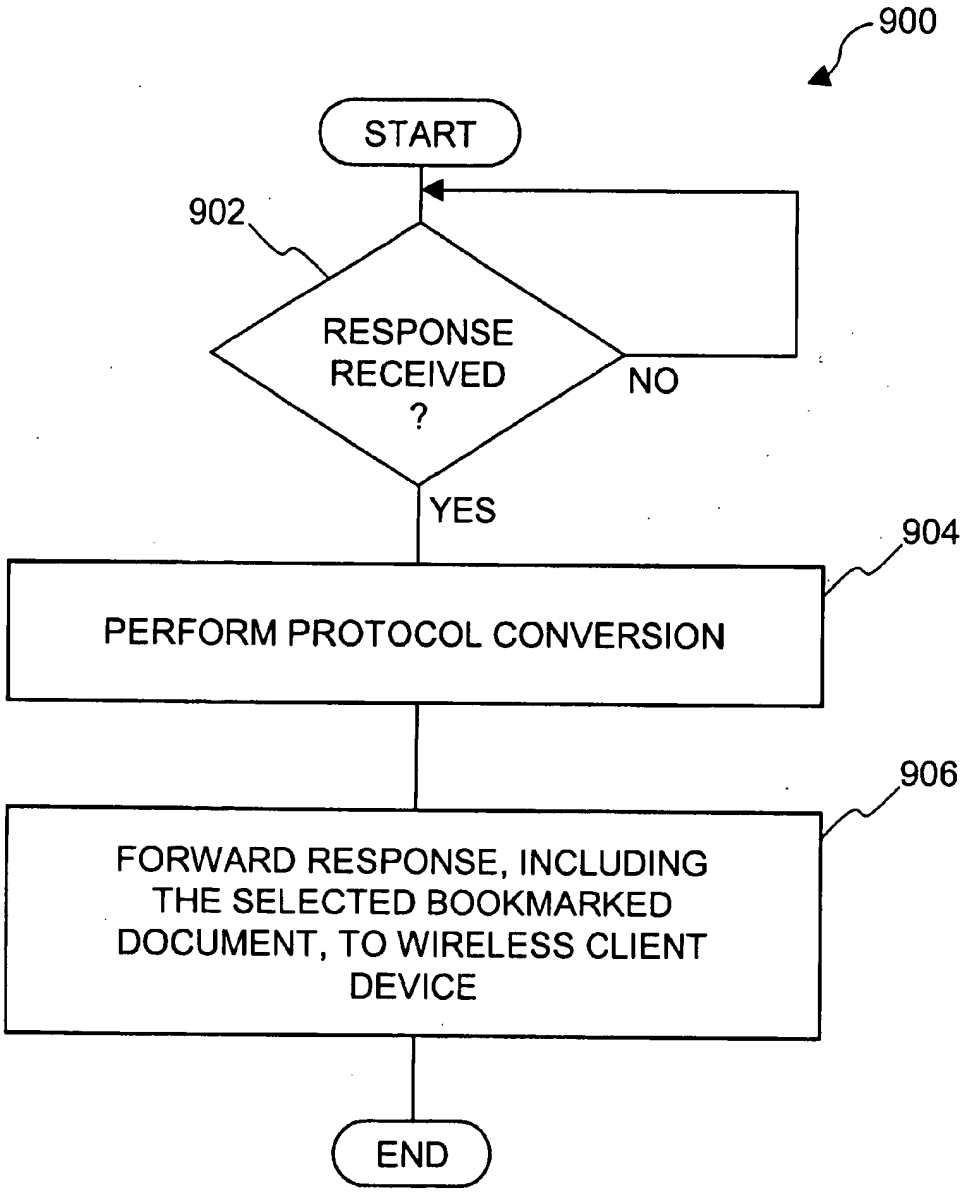

900
START
902
RESPONSE RECEIVED ?
NO
YES
904
PERFORM PROTOCOL CONVERSION
906
FORWARD RESPONSE, INCLUDING THE SELECTED BOOKMARKED DOCUMENT, TO WIRELESS CLIENT DEVICE
END

Fig. 9

REMOTE BOOKMARKING FOR WIRELESS CLIENT DEVICES

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates to wireless client devices and, more particularly, to a remote bookmarking for wireless client devices.

2. Description of the Related Art

The explosion of hypertext based technologies has spread to the domain of wireless communication systems. Wireless client devices (e.g., two-way pagers, cellular phones, palm sized computing devices and personal digital assistants (PDAs)) and wireless network protocols have been designed which permit interactive access to remote information services (e.g., directory services, financial information, new services, sport information and traffic & weather reports) through a variety of wireless and wired networks, most notably the World Wide Web (WWW) service of the Internet.

One aspect of this technology that has lagged behind the rest is that of input technology and data entry. The primary method of data entry for most wireless client devices is by a keypad. Due to size constraints and cost considerations, the keypads of these wireless client devices are particularly cumbersome. Keypads usually have between 16 and 24 keys, which is a sufficient number for numeric input but very inefficient when dealing with a standard alphanumeric data set (i.e., ASCII).

A user requesting information from the Internet accesses information servers using a web browser. For example, a user requesting information on stock purchases might input the following string:

http://www.stocktips.net/

Individual web addresses of this type is easily entered in an address box of a web browser used with standard desktop and laptop computers. Such web browsers can, for example, include Netscape Navigator from Netscape Communication Corporation or Microsoft Explorer from Microsoft Corporation. However, most users have a plurality of such sites which are frequently revisited. For the user's convenience, web browsers have also provided a bookmark function that allows a user to save the web addresses of those web sites that the user desires to return to in the future.

Bookmarks for a web browser operate very similar in function to the bookmarks used to hold a place (a page or section of interest) in a book. In the case of the web browser, the bookmark is electronic and the place being held is an electronic document (e.g., a web page) located on an information server and particularly identified by a uniform resource locator (URL). A user bookmarks a web page of interest using a menu selection provided by a user interface of the web browser. For example, Microsoft Explorer has a "favorites" menu that allows a user to provide a bookmark (and a title or label for the bookmark) to the web page currently being viewed by the web browser.

Bookmarks offer two very significant conveniences: 1) Bookmarks free users from having to remember or write down uniform resource locators (URLs) for information sites of interest; and 2) Bookmarks significantly increase information site access speeds by minimizing navigation actions with the user interface. In today's fast paced techno-environment, such conveniences have become very important for service providers and users alike.

Acceptance of wireless client devices with network access (e.g., Internet and intranets) will be significantly improved if users can be presented with a user interface that helps to minimize the limitations (e.g., limited processing/memory capabilities and a cumbersome user interface) of these devices. The use of bookmarks in conjunction with these devices will represent a significant advance in the desired direction.

The existing approaches to use of bookmarks on wireless client devices have various problems. One problem with existing approaches is that several navigation actions are required to select a single bookmark. Each navigation action is time consuming and leads to user dissatisfaction. Another problem associated with using bookmarks in conjunction with wireless client devices is that transmission efficiency when using bookmarks is low because URLs, which can be lengthy, need to be transmitted from the wireless client devices in requests for the bookmarked documents identified by the URLs. Still another problem is that with existing approaches to bookmarks, the URLs for the bookmarks need to be provided on or obtained by and stored in the wireless client device seeking to make use of a bookmark. Although storage of bookmarks and their associated URLs on desktop and laptop computers does not significantly impact the memory resources of desktop and laptop computers, storage of bookmarks and their associated URLs does represent a significant burden on the limited memory resources of wireless client devices. Bandwidth requirements for transmissions over wireless networks is a primary consideration for both service provider and user alike. Some of the URLs can be rather lengthy, requiring considerable bandwidth and airtime.

Thus, there is a need for improved approaches to enable a wireless client device to implement bookmarks with improved transmission efficiency, less navigation actions and/or reduced amounts of memory resources.

SUMMARY OF THE INVENTION

Broadly speaking, the invention relates to improved techniques that enable wireless devices to implement bookmarks with improved transmission efficiency, reduced user navigation and/or reduced amounts of memory resources. One aspect of the invention pertains to use of a compact request from a wireless device to an intermediate server when requesting a document or file by selection of a bookmark. Another aspect of the invention is the ability of a user to select a bookmark to request the associated document or file with reduced user interaction (e.g., a single button action). Still another aspect of the invention is that memory resources of the wireless devices need not be consumed to store network addresses (e.g., URLs) for the bookmarks. These aspects and other aspects described below can be used separately or in combination.

The invention can be implemented in numerous ways, including as a method, an apparatus, a system, and a computer readable medium. Several embodiments of the invention are discussed below.

As a method for requesting a document on a remote server using a user interface of a wireless client device, one embodiment of the invention includes the operations of obtaining a compact bookmark identifier for a selected bookmarked document, and transmitting a compact request including the compact bookmark identifier to a translation server. Neither the compact bookmark identifier nor the compact request include a universal resource locator for the selected bookmarked document.

As a method for utilizing bookmarks on a wireless client device, one embodiment of the invention includes the opera-

tions of: selecting one of a plurality of bookmarks available to the wireless client device; transmitting a compact request for the document or file represented by the selected bookmark from the wireless client device to an intermediate server; obtaining, from the intermediate server, a universal resource locator for the document or file represented by the selected bookmark; preparing a non-compact request for the document or file represented by the selected bookmark, the non-compact request including the universal resource locator for the document or file represented by the selected bookmark; and forwarding the non-compact request for the document or file represented by the selected bookmark to a remote server identified by at least a portion of the universal resource locator.

As a method of selecting among a plurality of information servers by a wireless client device having a display and a keypad, one embodiment of the invention includes the operations of: providing a dual function key on the keypad of the wireless client device, the dual function key having a primary function and a secondary function; executing the primary function if the dual function key is pressed for less than a predetermined time period; executing the secondary function if the dual function key is pressed for a time period greater than or equal to the predetermined time period, the execution of the secondary function producing a compact bookmark request; and forwarding the compact bookmark request to an intermediate server device over a wireless network using a first communications protocol. The compact bookmark request is used to access bookmark information available to the intermediate server device to produce a request to one of the plurality of information servers that couple to the intermediate server device through a wired network using a second communications protocol.

As a computer readable medium including computer program code for requesting a page on a remote server using a user interface of a wireless client device, one embodiment of the invention includes: computer program code for obtaining a compact bookmark identifier for a selected bookmarked page; computer program code for producing a compact request for the selected bookmarked page, the compact request including the compact bookmark identifier and not including a universal resource locator for the selected bookmarked page; and computer program code for transmitting a compact request to a translation server.

As a computer readable medium for utilizing bookmarks on a wireless client device, one embodiment of the invention includes: computer program code for selecting one of a plurality of bookmarks available to the wireless client device; computer program code for transmitting a compact request for the document or file represented by the selected bookmark from the wireless client device to an intermediate server; computer program code for obtaining, from the intermediate server, a universal resource locator for the document or file represented by the selected bookmark; computer program code for preparing a non-compact request for the document or file represented by the selected bookmark, the non-compact request including the universal resource locator for the document or file represented by the selected bookmark; and computer program code for forwarding the non-compact request for the document or file represented by the selected bookmark to a remote server identified by at least a portion of the universal resource locator.

As a wireless communication system, one embodiment of the invention includes a plurality of wireless client devices and a server device coupled to a wireless network servicing the wireless client devices. Each of the wireless devices including a keypad, a memory, a screen display and a processor. The processor operates to execute computer program code to generate a compact bookmark when a key on the keypad is depressed and held for a predetermined time period. The server device provides storage for bookmark information for the wireless client devices. The bookmark information is stored associated with user accounts for the wireless communication devices. Upon receiving a compact bookmark from one of the wireless communication devices, the server produces a request to a remote server storing a document or file associated with the compact bookmark, where the request is formed based on the compact bookmark and bookmark information.

The advantages of the invention are numerous. Different embodiments or implementations may yield one or more of the following advantages. One advantage of the invention is that bookmarks are able to be selected with greater speed and ease. Another advantage of the invention is that between the wireless client device and an intermediate server (e.g., proxy server) a compact request format is used so as to substantially reduce the amount of data to be transmitted for a request. Still another advantage of the invention is that memory storage at the wireless client device to support bookmarks is reduced.

Other aspects and advantages of the invention will become apparent from the following detailed description taken in conjunction with the accompanying drawings which illustrate, by way of example, the principles of the invention.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will be readily understood by the following detailed description in conjunction with the accompanying drawings, wherein like reference numerals designate like structural elements, and in which:

FIG. **1** illustrates a schematic configuration in which the present invention may be practiced;

FIG. **2** illustrates a functional block diagram of a wireless client device according to an embodiment of the present invention;

FIG. **3** illustrates a functional block diagram of a proxy server device according to an embodiment of the present invention;

FIG. **4** illustrates a "Keypad Bookmark Manager" used to assign the bookmarks to the keys of the wireless client device according to one embodiment of the present invention;

FIG. **5** illustrates the user interface and display of a wireless client device according to one embodiment of the present invention;

FIG. **6** is a flowchart of the bookmark processing according to one embodiment;

FIG. **7** is a flow diagram of client-side request processing according to one embodiment of the invention;

FIG. **8** is a flow diagram of intermediate request processing according to one embodiment of the invention; and

FIG. **9** is a partial flow diagram of intermediate server response processing according to one embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

In the following detailed description of the present invention, numerous specific details are set forth in order to provide a thorough understanding of the present invention.

However, it will become obvious to those skilled in the art that the present invention may be practiced without these specific details. In other instances, well known methods, procedures, components, and circuitry have not been described in detail to avoid unnecessarily obscuring aspects of the present invention.

The detailed description of the present invention in the following are often presented in terms of procedures, steps, logic blocks, processing, and other symbolic representations that resemble of data processing devices coupled to networks. These process descriptions and representations are the means used by those experienced or skilled in the art to most effectively convey the substance of their work to others skilled in the art.

The invention relates to improved techniques that enable wireless devices to implement bookmarks with improved transmission efficiency, reduced user navigation and/or reduced amounts of memory resources. One aspect of the invention pertains to use of a compact request from a wireless device to an intermediate server when requesting a document or file by selection of a bookmark. Another aspect of the invention is the ability of a user to select a bookmark to request the associated document or file with reduced user interaction (e.g., a single button action). Still another aspect of the invention is that memory resources of the wireless devices need not be consumed to store network addresses (e.g., URLs) for the bookmarks. These aspects and other aspects described below can be used separately or in combination.

Wireless client devices, also referred to as two-way interactive communication or mobile devices, include but are not limited to personal digital assistant (PDA) like devices, cellular phones, or wireless capable remote controllers. Such devices typically have significantly less memory and processing capability than is found in desktop and laptop computers. These wireless client devices, which are not a combination of a computer and a wireless communications module, have a small display screen and a limited keypad as opposed to the keyboards associated with desktop computers.

FIG. **1** is a block diagram of an information retrieval system according to one embodiment of the invention. The information retrieval system allows a plurality of two-way wireless interactive communication devices **100** to information from remote information servers. The plurality of two-way wireless interactive communication devices **100**, referred to as wireless client devices or mobile devices herein, are serviced by airnet **112**. Although only one two-way wireless interactive communication devices **100** is shown in the FIG. **1**, the information retrieval system supports many two-way wireless interactive communication devices **100**. More generally, the airnet **112** is a wireless network and can be implemented in a variety of types of wireless networks. Examples of commonly used wireless networks include Cellular Digital Packet Data (CDPD), Global System for Mobile Communications (GSM), Code Division Multiple Access (CDMA) and Time Division Multiple Access (TDMA), to name a few.

Also shown in FIG. **1** is landnet **120**, which is a landline network. More generally, landnet **120** is a wired network. For example, landnet **120** may be the Internet, an intranet or other data network. Coupled to landnet **120** are a personal computer (PC) **128** and an information server device **124**. Personal computer **128** may be a commonly available laptop or desktop computer and run a HyperText Markup Language (HTML) browser, such as Netscape Navigator from Netscape Communications Corporation (www.netscape.com). The HTML browser can communicate with the information server device **124** via landnet **120** using HyperText Transfer Protocol (HTTP) to thus access information stored in the information server device **124**. As an example, the information server device **124** is a workstation computer such as available from SUN Microsystems Inc. (www.sun.com). The information stored in the information server device **124** may be hypermedia information. Although not shown, various other servers or computers are connected to landnet **120**.

A proxy server device **116** is coupled between landnet **120** and airnet **112**. The proxy server device **116** is also known as a network gateway server. The proxy server device **116** can be implemented as a workstation computer or a personal computer. Often, the communication protocol used in airnet **112** is different from that used in landnet **120**. As a result, one of the functions that proxy server device **116** performs is to map or translate from one communication protocol to another, thereby wireless device **100** coupled to airnet **112** can communicate with any of the information server devices (e.g. information server **124**) coupled to landnet **120** via proxy server device **116**. The proxy server device **116** also store or provide access to accounting services, configuration management services, and dedicated storage for applications and files for user accounts. These applications and services may be resident on proxy server device **116** or on a separate server device accessible via landnet **120**.

According to one embodiment, the communication protocol used by information server device **124** is the well known HyperText Transfer Protocol (HTTP) or its secure version (HTTPS). HTTP operates on Transmission Control Protocol (TCP) and controls the interaction of personal computer **128** and information server **124** with landnet **120**. In such an embodiment, information server **124** operates as a HTTP server and personal computer **128** operates a HTML browser.

The communication protocol between wireless client device **100** and proxy server device **116** via airnet **112** is, for example, Handheld Device Transport Protocol (HDTP) (formerly known as Secure Uplink Gateway Protocol (SUGP)) or Wireless Access Protocol (WAP). HDTP and WAP can run on User Datagram Protocol (UDP). In such an embodiment, the interaction between proxy server device **116** and wireless client device **100** uses HDTP over UDP. The wireless client device **100** operates a Handheld Device Markup Language (HDML) browser (also known as a micro-browser).

It should be noted that HDTP is a session-level protocol that resembles HTTP but without incurring the overhead thereof and is highly optimized for use in thin devices, such as mobile devices that have significantly less computing power and memory than a desktop personal computer. Further, it is understood to those skilled in the art that UDP does not require a connection to be established between a client and a server before information can be exchanged, which eliminates the need of exchanging a large number of packets during a session creation between a client and a server. Exchanging a very small number of packets during a transaction is a desired feature for a mobile device with very limited computing power and memory to effectively interact with a landline device.

HDML is a tag based document language and comprises a set of commands or statements specified in a card that specifies how information displayed on a small screen **104** of the wireless client device **100**. Normally, a number of

cards are grouped into a deck that is a unit of HDML information exchanged between wireless client device **100** and proxy server device **116**. The specifications of HDTP, entitled "HDTP Specification" and HDML, entitled "HDML 2.0 Language Reference" are incorporated herein by reference in their entirety.

The communication protocols (HDTP, HTTP, and HTTPs) and the markup languages (HDML and HTML) are presented for purposes of illustration and not limitation. One skilled in the art will appreciate that the present invention can be practiced using other communications protocols (e.g., Wireless Session Protocol (WSP)) and markup languages (e.g., Compact Hypertext Markup Language (cHTML) and Wireless Markup Language (WML).

Some of the features in wireless client device **100** that make the disclosed system work more efficiently are described below. According to one embodiment, wireless client device **100** includes a display screen **104** and a phone keypad **108** which allow a user thereof to interact with wireless client device **100**. Phone keypad **108** preferably provides a typical phone keypad, a pair of generic buttons, and at least a pair navigation buttons. The typical phone keypad, as commonly seen, has twelve buttons. Of the twelve buttons, ten buttons are consecutively numbered (0 to 9), one button is designated "*", and the other button is designated "#". Further, it is to be understood by those of ordinary skill in the art that the present invention may be practiced using various other types of input interfaces and arrangements (e.g., softkeys, iconic screens).

Further, there is a working memory (not shown) in wireless client device **100**. Compiled and linked processes are typically stored in the working memory as a client module that causes wireless client device **100** to interact with a remote server, such as proxy server device **116**. Upon activation of a predetermined key utilizing keypad **108**, for example, wireless client device **100** initiates a communication session with proxy server device **116** using the client module in the working memory. During the communication session, the wireless client device **100** requests certain information (e.g., a document) from information server **124** or proxy server device **116**. In one embodiment, in response to the request, wireless client device **100** typically receives a single HDML deck (from or through proxy server device **116** and stores (caches) the deck in the working memory. As described above, an HDML deck comprises one or more cards and each card includes the information required to generate a screen display on display screen **104**. The number of cards in a card deck can be selected to facilitate efficient use of the resources in wireless client device **100** and in airnet network **112**.

As used herein, a display screen is the physical display apparatus in a wireless client device, such as a 4-line by 20-character Liquid Crystal Display (LCD) screen. A screen display is an image presented or displayed on the display screen. Further, it is understood that a display screen having display lines is only for illustrative purpose and many display screens in reality are graphics-based and do not necessarily have distinct display lines and it will be appreciated that the principles of this invention are equally applicable thereto.

Although the information retrieval system illustrated in FIG. **1** includes the proxy server device **116**, it should be understood that the information retrieval system does not require that a proxy server device be present. Instead, the airnet **112** could couple the landnet **120** directly which would allow the wireless client device **100** to access information servers, such as the information server **124**.

According to one embodiment of the invention, wireless client device **100** and proxy server device **116** together provide bookmark features to the information retrieval system. A user of the wireless client device is able to utilize the bookmark features to gain rapid and easy access to previously bookmarked locations (e.g., remote information servers) or documents thereon. A bookmark is a shortcut feature that allows a user to access a remote location or document identified by a uniform resource locator (URL) without having to type in the URL. Hence, by selecting a bookmark, the user directs the wireless client device **100** to the remote location or document without having to perform the tedious data entry for the URL. According to the invention, the bookmarks for wireless client device **100** are stored in proxy server device **116** (or some other remote server coupled to landnet) and not normally stored in wireless client device **100**. As a result, the limited memory available on wireless client device is not consumed by storage of bookmarks and greater numbers of bookmarks can be supported. Since proxy server device **116** supports a large number of subscribers or wireless client devices, the bookmarks for each subscriber or wireless client device are stored separately along with other information (e.g., account, configuration, and preference information).

Bookmarks for a subscriber or wireless client device **100** can be entered using personal computer **128** or wireless client device **100**. For example, personal computer **128** can access to proxy server device **116** through the landnet using, for example, a web browser to edit, create and delete bookmarks for the wireless client device **100**. Alternatively, a user of the wireless client device **100** can interact with the keypad **108** and other buttons or input areas to edit, create and delete bookmarks for the wireless client device **100**. When a bookmark is created it can also be provided with a shortened name that can be displayed on display screen **104** as needed or requested to remind the user of the previously assigned bookmarks.

According to one embodiment of the invention, the bookmarks are assigned to particular keys on keypad **108** of wireless client device **100**. A user wishing to access information server **124** (e.g., a document or file on information server **124**) from wireless client device **100** using a previously defined bookmark need only press and hold the assigned key. Pressing and holding the assigned key causes wireless client device **100** to generate a compact bookmark identifier which is transmitted to proxy server device **116** in a compact request for the bookmarked document (or file). The proxy server device **116** will intercept the compact request from the wireless client device **116** and convert the compact request into a normal request. Namely, the compact request using the compact bookmark identifier, whereas the normal request uses the appropriate URL previously associated with the bookmark. The appropriate URL is stored at proxy server device **116** (or accessible thereto) and is associated with a device identifier associated with wireless client device **100** or a subscriber identifier associated with the subscriber.

FIG. **2** is a block diagram of an information retrieval system according to another embodiment of the invention. The information retrieval system includes wireless client device **200**, personal computer **270**, proxy server device **254**, and information server **280**. Proxy server device **254**, information server **280** and personal computer **270** respectively correspond to proxy server device **116**, information server **124**, and personal computer **128** of FIG. **1**. The wireless client device **200** is, for example, a detailed embodiment of wireless client device **100**. To avoid obscur-

ing the principle aspects of the present invention, well known methods, procedures, components and circuitry in wireless client device **200** are not described in detail.

Wireless client device **200** includes a Wireless Control Protocol (WCP) interface **236** that couples to airnet **250** via a RF transceiver (not shown) to receive incoming and outgoing data signals. Device identifier (ID) storage **208** supplies a device ID to WCP interface **236**. The device ID identifies a specific code that is associated with wireless client device **200** and directly corresponds to the device ID in a subscriber (user) account provided in proxy server device **254**. In addition, wireless client device **200** includes a client module **212** with an interface engine **216** which works in conjunction with processor **204** and working memory **240** to perform the processing tasks performed by wireless client device **200** including establishing a communication session with proxy server device **254** via airnet **250**, requesting and receiving data via airnet **250**, displaying information on a display screen through the use of display circuitry **220**, and receiving user input from a user via a keypad controlled by keypad circuit **224**. Additionally, the client module **212** operates, among other things, a browser, commonly referred to as micro-browser, requiring much less computing power and memory than well-known HTML browsers do. The micro-browser is, for example, a HDML micro-browser available from Unwired Planet, Inc. located at 800 Chesapeake Drive, Redwood City, Calif. 94063. Additional details on accessing a (proxy) server device from a wireless client device using a (micro) browser is described in U.S. patent application Ser. No. 08/570,210, now U.S. Pat. No. 5,809,415, which is hereby incorporated by reference in its entirety.

Wireless client device **200** also includes the voice circuitry **232** (e.g., a speaker and a microphone) and the associated hardware (e.g., encoder/decoder **228**, processor **204** and keypad circuitry **224**) which allows it to switch to a telephone mode of operation which is separate and distinct from a network (data) mode of operation used when interfacing with proxy server device **254** and other devices on landnet **260**.

According to one embodiment of the present invention, a user desiring to obtain information from information server **280**, places wireless client device **200** in the network mode of operation and presses and holds a pre-assigned key on the keypad of wireless client device **200**. The software stored in client module **212** causes the key that was pressed and held to be recognized in a unique manner than that same key would be recognized in the telephone mode of operation. Specifically, instead of generating a alphanumeric character (e.g. "1"), a compact bookmark identifier is generated. Wireless client device **200** establishes a connection with proxy server device **254** via airnet **250** and transmits a compact request for the document identified by the compact bookmark identifier. The compact bookmark identifier is used to access previously stored bookmark information on proxy server device **254**. Using the stored bookmark information, proxy server device **254** generates a normal request for the document originally identified by the compact bookmark identifier. The normal request is then forwarded over landnet **260** to information server **280** (where the requested documents resides). One of ordinary skill in the art will appreciate that if wireless client device **200** is in a non-network mode (e.g., telephone mode) of operation then software within wireless client device **200** can cause wireless client device **200** to be placed in the network mode of operation upon detecting that a pre-assigned key has been press and held so as to select a previously assigned bookmark.

In the case where wireless client device **200** uses HDML or WML protocols, various operations on wireless client device are controlled or provided through card decks. Card decks contain one or more cards of HDML or WML documents. For example, an interface card deck can be provided on wireless client device **200** to facilitate a user creating, modifying or deleting bookmarks. According to one embodiment, the bookmarks are assigned to the keys associated with the keypad on wireless client device **200**. It is important to note that any key on the keypad of wireless client device **200** can be assigned as a bookmark. When bookmarks are created, modified or deleted using wireless client device **200**, an interface card deck can control the user interface provided to the user via the display screen. As noted above, the bookmarks can also be created, modified or deleted through use of personal computer **270**.

More particularly, when needed, an interface card deck is received by wireless client device **200** and is loaded into working memory **240**. The interface card deck is then processed by processor **204** and client module **212** to produce a user interface on the screen display. The interface card deck is comprised of one or more markup language entities, which contribute to the functionality of the user interface on wireless client device **200**. The functions provided include: 1) information display; 2) list selection; 3) input operations; and 4) control functions. These functions will enable the user interface of the wireless client device **200** to function more efficiently during bookmark creation, modification, and deletion. The interface card deck is navigated using the previously described microbrowser. For example, the interface card deck can be used to prompt the user for input (e.g., "Address for the Site is?", "Name for the Site?", "Press the Key You Want to Assign").

Additionally, it will be appreciated by one of ordinary skill in the art that this method of bookmark assignment and utilization may be practiced using user interfaces other than keypads (e.g., iconic interfaces, generic buttons, special buttons, soft buttons). Further, although the selection of a previously established bookmark is achieved through a press and hold of a pre-assigned key, the selection of a bookmark can be achieved in other ways, including a double-click of a button, a short sequence of buttons, etc.

It should also be noted that an interface card deck can also display a list of bookmarks that have already been assigned. As an example, the list of bookmarks that have already been assigned can be provided to and displayed by the wireless client device by pressing and holding a predetermined key (e.g., "0") on the keypad. A user could then navigate through the list to select a bookmark from the list. This type of selection, however, does not offer the advantages of minimal user actions to obtain a selection as does the simple selection of a pre-assigned button.

FIG. **3** is a detailed block diagram of a proxy server device **300** according to one embodiment of the invention. Proxy server device **300** comprises a server module **308** coupled between LCP interface **348** and WCP interface **304**. Server module **308**, which is typically loaded in memory, performs traditional server processing as well as protocol conversion processing from one communication protocol to another communication protocol. More particularly, server module **308** is coupled to a landnet **352**, which uses a first communication protocol (e.g., Hypertext Transfer protocol (HTTP) or Secure Hypertext Transfer Protocol (HTTPS)), and to an airnet **310** which uses a second communication protocol (e.g., Handheld Device Transport Protocol (HDTP) or Wireless Access Protocol (WAP)).

It is understood to those skilled in the art that a server device used herein, which may perform as proxy server

device **254** and be coupled to landnet **260**, refers to a piece of hardware equipment that comprises one or more microprocessors, working memory, buses and necessary interfaces and other components. One the other hand a server module refers to compiled and linked processes of the disclosed system loaded into the working memory to perform designated functions through the parts and components in the server device.

Server module **308** comprises a control engine **312**, a message processor **324**, an account manager **316**, and an account interface **320**. Control engine **312** interacts with the client module of wireless client device (not shown) through airnet **310** and coordinates the reception of requests. Message processor **324** receives messages from landnet **352** and performs a series of processing and management activities. The processing performed by message processor **324** includes protocol conversion between the different protocols used on airnet **310** and landnet **352**.

Account manager **316** manages through account interface **320** a number of subscriber (user) accounts for all the wireless client devices serviced by proxy server device **300**. Each of the wireless client devices serviced by proxy server device **300** is assigned a device identifier (ID). Device ID can be a phone number of the device or an IP address or a combination of an IP address and a port number, for example: 204.163.165.132:01905 where 204.163.165.132 is the IP address and 01905 is the port number. The device ID is further associated with a subscriber ID created and administrated by a carrier and stored in proxy server device **300** as part of the procedures to activate a subscriber account for a wireless client device. The subscriber ID may take the form of, for example, 861234567-10900_pn.mobile.att.net by AT&T Wireless Service, and is a unique identification to a wireless client device.

Upon receiving a compact request having a compact bookmark identifier for a previously assigned bookmark, proxy server device **300** accesses the subscriber account (corresponding to the subscriber identification number of the wireless client device that sent the compact bookmark identifier) contained within proxy server device **300** or in a remote server accessed via landnet **352**. The subscriber account contains bookmark information that has been previously stored. The bookmark information includes a Uniform Resource Locator (URL) for the selected bookmark being identified by the compact bookmark identifier. The bookmark information can also include a short name for the bookmark. Proxy server device **300** can also forward a wireless client device a interface card deck for a menu list of previously assigned keys that can be displayed on the display screen of the requesting wireless client device so that a user can determine which keys are assigned to which bookmarks. In such case, the short names serve to inform the user of the location or document of the bookmark. One example of a short name is "Acme" for the full name "Acme Corporation—Home Page". Such short names are more likely able to fit on the limited size screen display and more likely to be more descriptive of the bookmark.

FIG. **4** is an exemplary screen shot **400** for a Keypad Bookmark Manager according to one embodiment of the invention. A user wishing to create, modify or delete bookmarks for a wireless client device (e.g., wireless client device **100** of FIG. **1**) can access Keypad Bookmark Manager using a computer (e.g., computer **128** of FIG. **1**) with network connectivity (e.g., the Internet) and a web browser (e.g., Netscape Navigator). Each user or subscriber to wireless network service can be given a personal home page which they can visit to access Keyboard Bookmark Manager.

In the screen shot **400** for Keypad Bookmark Manager an iconic symbol **404** representing a "0" key shows that this key has been previously assigned to a location (e.g., document or web page address) having a specific URL **412**. The specific URL in this example is: http):// www.uplanet.bookmarks.smethers.com. The short name for the bookmark assigned to the "0" key is "Keypad Menu" as shown in field **408**. In this example, the fields in **408** and **412** are pre-set and are not normally able to be modified. The other fields displayed in the screen shot **400** of Keypad Bookmark Manager may be assigned by the user. The screen shot **400** shows that bookmarks for keys "1", "2" and "3" have been assigned by the user, and that key "4" is as yet unassigned. With respect to key "1", the screen shot **400** show that the bookmark has a short name of"My Stocks" in field **420** and a URL of "http://www.uplanet.com/stocks.html/" in filed **424**. Keys "2" and "3" are also shown in the screen shot **400** has having been assigned in accordance with information in fields **436, 440, 452** and **456**. When a user first enters a bookmark or modifies a bookmark, the entry is registered by activating the "SUBMIT" button associated with the assigned key (see iconic buttons **428, 444, 460** and **476**). For example, upon entering a bookmark assignment for the "2" key "SUBMIT" button **444** would be pressed. Fields **468** and **472** for the "4" key are current unassigned and available to store the user's next bookmark. Although only numbered keys have been described in this example, it would be apparent to one of ordinary skill in the art that any key on the keypad or other button or input selection mechanism of the subject wireless client device may be similarly assigned.

FIG. **5** is an exemplary wireless client device **500** for use with the invention. Once the bookmarks for wireless client device **500** have been assigned, pressing and holding an assigned key while the device is in the network (data) mode of operation will cause a compact request including the compact bookmark identifier to be obtained and forwarded to the associated proxy server device. For example, pressing and holding the "0" key will cause the compact bookmark identifier for that key to be generated and forwarded to the proxy server device in a compact request. The compact bookmark identifier is preferably two bytes in size. For example, the compact bookmark identifier can be two characters, a control character indicating a bookmark and a number for the particular bookmark. In the case of the "0" key and the assignments shown in FIG. **4**, the proxy server device retrieves the bookmark associated with the "0" key (i.e., http://www.uplanet.bookmarks.com/smethers.html and forwards a request for the page identified by the retrieved URL. The information server containing that document responds to the request and forwards the requested document or file to the wireless client device. In this example, the retrieved URL happens to address a HDML file residing on the proxy server device. This HDML file contains a listing the assigned keys to bookmarks and the associated short names that can be displayed on the screen display. Often, however, the URLs address documents or files on an information server located anywhere on the landnet. The wireless client device thereafter receives the requested document or file and displays information on the display screen **504**.

FIGS. **6** is a flowchart illustrating bookmark selection and processing **600** according to one embodiment of the present invention. The processing **600** begins at block **604** where a key press on a keypad is received. At block **608**, a determination is made as to whether or not the pressed key is a press and hold key. Here, in this embodiment, press and hold keys are those keys that are eligible for press and hold

entries. The press and hold are those keys that are eligible to be assigned to a bookmark. If the pressed key is not a press and hold key, then the pressed key will be processed as a normal input from an alpha-numeric numeric or function key at block **612**. On the other hand, if the pressed key is a press and hold key, then a determination is made as to whether the press key has also been held at block **616**. If the pressed key is determined not to have been held, then the processing **600** also performs block **612**. If, on the other hand, the pressed key was held, then the pressed key is processed as a bookmark at block **620**. In one embodiment, the pressed key is determined to be held is the key was pressed and held for a predetermined time period (e.g., 2 seconds). Hence, if the pressed key was pressed and released before the predetermined time period had expired, the pressed key is processed as a normal input from a alphanumeric or function key at block **612**. At block **620**, a bookmark request and a compact bookmark identifier will be generated and forwarded to the proxy server device storing the user's bookmark information. The proxy server device will use the compact bookmark identifier to retrieve the associated URL with the bookmark at block **624**. Then, a request will be generated and forwarded by the proxy server device to the information server identified by the URL at block **628**. Following block **628**, as well as following block **612**, the processing **600** is complete and ends.

FIG. **7** is a flow diagram of client-side request processing **700** according to one embodiment of the invention. The client-side request processing **700** is, for example, performed by the client module **212** of the wireless client device **200** illustrated in FIG. **2**.

The client-side request processing **700** begins with a decision block **702** that determines whether the user has requested a bookmarked document. Here, the client-side request processing **700** is essentially initiated when a user interacts with the wireless client device to select a bookmarked document that is to be requested. A user can request a bookmarked document in a variety of ways. In one embodiment, a bookmarked document is requested by the press and hold of a pre-assigned key of the wireless client device.

Once the decision block **702** has determined that the user has requested a bookmarked document, a compact bookmark identifier is obtained at block **704**. The compact bookmark identifier is a short identifier of at most a few bytes that identifies the particular bookmark that has been selected by the user. For example, for the wireless client device **500** illustrated in FIG. **5**, the telephone keypad **516** includes at least twelve buttons, namely buttons labeled 0–9, * and #. Each one of these at least twelve buttons can operate as a bookmark. As an example, when the user depresses button labeled "1", a first bookmark is selected and the compact bookmark identifier can be "&1". For such a compact bookmark identifier, only two bytes are need to identify the bookmark, a single bite for a bookmark control signal ("&") and another byte for a numeric value of the bookmark identifier ("1").

Next, a compact request is transmitted to a translation server. The compact request is a request for the bookmarked document that is sent to the translation server. However, the compact request is constructed such that to identify the bookmarked document, the compact bookmark identifier is contained within the compact request. The compact request will also, among other things, contain a device identifier (ID), a destination address and a source address.

At this point, the wireless client device is awaiting a response from the translation server. Hence, at block **708**, a decision block determines whether a response has been received. Once a response has been received, the response includes the selected bookmarked document that was previously requested. Hence, at block **710**, the selected bookmarked document causes information to be displayed on a screen display of the wireless client device. Following block **710**, the client-side request processing **700** is complete and ends.

FIG. **8** is a flow diagram of intermediate request processing **800** according to one embodiment of the invention. The intermediate request processing **800** is, for example, performed by the proxy server device **116** illustrated in FIG. **1**. Alternatively, the intermediate request processing can be performed by any other remote server coupled to the landnet (including the translation server of FIG. **7**).

The intermediate request processing **800** begins with a decision block **802** that determines whether a compact request has been received from a wireless client device. In other words, the intermediate request processing **800** is activated or begins when a compact request has been received from the wireless client device through the airnet. The compact nature of the compact request means that its size in terms of number of bytes is limited so that minimal bandwidth is consumed or needed and rapid transmission through the airnet **112** can be obtained. Once the decision block **802** determines that a compact request has been received, the compact bookmark identifier is extracted from the compact request at block **804**. The compact bookmark identifier is embedded within the compact request, typically as a field within the compact request. Hence, block **804** operates to parse the compact request to obtain the compact bookmark identifier.

Next, a URL associated with the compact bookmark identifier is obtained at block **806**. Here, for example, the remote server (translation server) stores a table for each subscriber supported by the airnet system. These tables store the URLs as associated with the subscriber's bookmarks as utilized on their wireless client devices. Upon receiving the compact request, a device identifier is obtained from the incoming compact request and used to obtain a subscriber ID which, in turn, identifies the table associated with the subscriber. Then, the compact bookmark identifier can be used to look-up within the table the appropriate URL (address) for the requested bookmarked document. Then, a standard request (i.e., no longer compact in nature) is prepared for the selected bookmarked document at block **808**. The standard request will, among other things, contain the obtained URL, the device identifier (ID), the destination address and the source address. Here, the remote server operates to form a traditional request using the URL that has been obtained from the table associated with the subscriber.

After the request is prepared, the request is transmitted to an information server at block **810**. However, in the case in which the intermediate request processing **800** is performed at the information server that is the destination for the requested bookmarked document, then block **810** may not be required. Following block **810**, the intermediate request processing **800** is complete and ends.

FIG. **9** is a partial flow diagram of intermediate server response processing **900** according to one embodiment of the invention. Often, the intermediate server will act as a network gateway or a proxy server for the wireless client device. Hence, in such cases, the response to the request that is transmitted to the information server (block **810**) is returned to the wireless client device through the intermediate server. Hence, the intermediate server response pro-

cessing **900** explains the return of the response from the information server through the intermediate server to the wireless client device. Namely, a decision block **902** determines whether the response has been received. If the response has not yet been received, the intermediate server response processing **900** is effectively waiting to receive the response. Once the response has been received, the intermediate server performs protocol conversion at block **904**. For example, the protocol conversion can be from HTML to HDML. Then, the response is forwarded to the wireless client device at block **906**. The response being forwarded to the wireless client device includes the selected bookmarked document that was originally requested by the wireless client device using the selection of a previously defined bookmark. Following block **906**, the intermediate server response processing **900** is complete and ends.

The invention can also be embodied as computer readable code on a computer readable medium. The computer readable medium is any data storage device that can store data which can be thereafter be read by a computer system. Examples of the computer readable medium include read-only memory, random-access memory, CD-ROMs, magnetic tape, optical data storage devices. The computer readable medium can also be distributed over a network coupled computer systems so that the computer readable code is stored and executed in a distributed fashion.

It can be appreciated by now that the present invention provides an efficient means for users of wireless client devices to navigate information services over a network using bookmarks stored on a remote server. This invention allows users to navigate such information services with a minimal amount of input interaction (i.e., key strokes) and with efficient use of the wireless client devices bandwidth.

The advantages of the invention are numerous. Different embodiments or implementations may yield one or more of the following advantages. One advantage of the invention is that bookmarks are able to be selected with greater speed and ease. Another advantage of the invention is that between the wireless client device and an intermediate server (e.g., proxy server) a compact request format is used so as to substantially reduce the amount of data to be transmitted for a request. Still another advantage of the invention is that memory storage at the wireless client device to support bookmarks is reduced.

The present invention has been described in sufficient detail with a certain degree of particularity. It is understood to those skilled in the art that the present disclosure of embodiments has been made by way of example only and that numerous changes in the arrangement and combination of parts as well as operations may be resorted without departing from the spirit and scope of the invention as claimed. Accordingly, the scope of the present invention is defined by the appended claims rather than the forgoing description of embodiments.

What is claimed is:

1. A method for requesting a document on a remote server using a user interface of a wireless client device, the method comprising:

obtaining a compact bookmark identifier for a selected bookmarked document without keying in the compact bookmark identifier at the wireless client device when initiating a request for the selected bookmarked document; and

transmitting a compact request including the compact bookmark identifier to a translation server,

wherein the selected bookmarked document is chosen by a selection of a single button on the wireless client device, and

wherein neither the compact bookmark identifier nor the compact request include a universal resource locator for the selected bookmarked document.

2. A method as recited in claim **1**, wherein the selected bookmarked document is a web page.

3. A method as recited in claim **1**, wherein the translation server is a proxy server.

4. A method as recited in claim **3**, wherein the selection of the button is determined by depressing the button and holding the button depressed for at least a predetermined period of time.

5. A method as recited in claim **1**, wherein the wireless client device includes a screen display,

wherein the selected bookmarked document is further chosen without needing to first display a list of available bookmarks on the screen display of the wireless client device.

6. A method for utilizing bookmarks on a wireless client device, the method comprising:

selecting one of a plurality of bookmarks available to the wireless client device by selection of a single button on the wireless client device;

transmitting a compact request for the document or file represented by the selected bookmark from the wireless client device to an intermediate server;

obtaining, from the intermediate server, a universal resource locator for the document or file represented by the selected bookmark;

preparing a non-compact request for the document or file represented by the selected bookmark, the non-compact request including the universal resource locator for the document or file represented by the selected bookmark; and

forwarding the non-compact request for the document or file represented by the selected bookmark to a remote server identified by at least a portion o the universal resource locator.

7. A method as recited in claim **6**, wherein the document or file represented by the selected bookmark is a web page.

8. A method as recited in claim **7**, wherein the intermediate server is a proxy server.

9. A method as recited in claim **6**, wherein the selection of the button is determined by depressing the button and holding the button depressed for at least a predetermined period of time.

10. A method as recited in claim **9**, wherein the wireless client device includes a screen display,

wherein the selecting one of the plurality of bookmarks available to the wireless client device is performed by selection of a button on the wireless client device without needing to first display a list of available bookmarks on the screen display of the wireless client device.

11. A method as recited in claim **6**, wherein the compact request being transmitted from the wireless client device to an intermediate server does not include the universal resource locator for the selected bookmark.

12. A method as recited in claim **6**, wherein the obtaining of the universal resource locator comprises:

obtaining an identifier for the wireless client device or its subscriber from the compact request; and

locating bookmark information associated with the identifier at the intermediate server; and

retrieving, from the bookmark information, the universal resource locator for the document or file represented by the selected bookmark.

13. A method as recited in claim 12, wherein the selecting one of the plurality of bookmarks available to the wireless client device is performed by selection of a single button on the wireless client device.

14. A method as recited in claim 13, wherein the selection of the button is determined by depressing the button and holding the button depressed for at least a predetermined period of time.

15. In a wireless client device having a display and a keypad, a method of selecting among a plurality of information servers, the method comprising:

providing a dual function key on the keypad of the wireless client device, the dual function key having a primary function and a secondary function;

executing the primary function if the dual function key is pressed for less than a predetermined time period;

executing the secondary function if the dual function key is pressed for a time period greater than or equal to the predetermined time period, the execution of the secondary function producing a compact bookmark request; and

forwarding the compact bookmark request to an intermediate server device over a wireless network using a first communications protocol,

wherein the compact bookmark request is used to access bookmark information available to the intermediate server device to produce a request to one of the plurality of information servers that couple to the intermediate server device through a wired network using a second communications protocol, and

wherein said executing of the secondary function and said forwarding of the compact bookmark request are performed in response to a single press of the dual function key.

16. A method as recited in claim 15, wherein the first communications protocol for the wireless network is a wireless communications protocol and the second communications protocol for the wired network is Hypertext Transport Protocol (HTTP) over Internet Protocol (TCP/IP).

17. A method as recited in claim 15, wherein the wireless client device is selected from a group consisting of mobile telephones, pagers and Personal Digital Assistants having screen displays.

18. A computer readable medium including computer program code for requesting a page on a remote server using a user interface of a wireless client device, the computer readable medium comprising:

computer program code for obtaining a compact bookmark identifier for a selected bookmarked page without the user keying in the compact bookmark identifier at the wireless client device when initiating a request for the selected bookmarked document;

computer program code for detecting selection of a single button on the wireless client device, thereby selecting the selected bookmarked document;

computer program code for producing a compact request for the selected bookmarked page, the compact request including the compact bookmark identifier and not including a universal resource locator for the selected bookmarked page; and

computer program code for transmitting a compact request to a translation server.

19. A computer readable medium as recited in claim 18, wherein the computer program code for detecting selection of the button operates to determine when a button on the wireless client device has been depressed and held depressed for at least a predetermined period of time.

20. A computer readable medium as recited in claim 19, wherein the translation server is a proxy server.

21. A computer readable medium including computer program code for utilizing bookmarks on a wireless client device, the computer readable medium comprising:

computer program code for selecting one of a plurality of bookmarks available to the wireless client device by selection of a single button on the wireless client device;

computer program code for transmitting a compact request for the document or file represented by the selected bookmark from the Wireless client device to an intermediate server;

computer program code for obtaining, from the intermediate server, a universal resource locator for the document or file represented by the selected bookmark;

computer program code for preparing a non-compact request for the document or file represented by the selected bookmark, the non-compact request including the universal resource locator for the document or file represented by the selected bookmark; and

computer program code for forwarding the non-compact request for the document or file represented by the selected bookmark to a remote server identified by at least a portion of the universal resource locator.

22. A computer readable medium as recited in claim 21, wherein the document or file represented by the selected bookmark is a web page.

23. A computer readable medium as recited in claim 21, wherein the intermediate server is a proxy server.

24. A computer readable medium as recited in claim 21, wherein the computer program code for selecting to detect the selection of the button by determining whether the button has been depressed and held depressed for at least a predetermined period of time.

25. A computer readable medium as recited in claim 24, wherein the wireless client device includes a screen display,

wherein the computer program code for selecting one of the plurality of bookmarks available to the wireless client device is performed by detecting a selection of the button on the wireless client device without needing to first display a list of available bookmarks on the screen display of the wireless client device.

26. A computer readable medium as recited in claim 21, wherein the compact request being transmitted from the wireless client device to an intermediate server does not include the universal resource locator for the selected bookmark.

27. A computer readable medium as recited in claim 21, wherein the computer readable medium for obtaining of the universal resource locator comprises:

computer program code for obtaining an identifier for the wireless client device or its subscriber from the compact request;

computer program code for locating bookmark information associated with the identifier at the intermediate server; and

computer program code for retrieving, from the bookmark information, the universal resource locator for the document or file represented by the selected bookmark.

28. A computer readable medium as recited in claim 27, wherein the computer program code for selecting one of the plurality of bookmarks available to the wireless client device operates to detect selection of a button on the wireless client device.

29. A computer readable medium as recited in claim 28, wherein the computer program code for selecting to detect the selection of the button by determining whether the button has been depressed and held depressed for at least a predetermined period of time.

30. A wireless communication system, the system comprising:

a plurality of wireless client devices, each of the wireless devices including a keypad, a memory, a screen display and a processor, the processor operates to execute computer program code to generate a compact bookmark when a single key on the keypad is depressed and held for a predetermined time period; and

a server device coupled to a wireless network servicing the wireless client devices, the server device provides storage for bookmark information for the wireless client devices, the bookmark information being stored associated with user accounts for the wireless communication devices, upon receiving a compact bookmark from one of the wireless communication devices the server produces a request to a remote server storing a document or file associated with the compact bookmark, the request being formed based on the compact bookmark and bookmark information.

31. A system as recited in claim 30,

wherein the server device is a proxy server device that couples between a wired data network and a wireless data network, and

wherein the wireless data network is capable of coupling to the wireless communication devices to facilitate communications between the proxy server and the wireless communication devices.

32. A system as recited in claim 31, wherein the wired data network uses a first communications protocol and the wireless data network uses a second communications protocol that differs from the first communication protocol.

33. A system as recited in claim 32, wherein the first communications protocol for the wireless data network is a wireless communications protocol and the second communications protocol for the wired data network is Hypertext Transport Protocol (HTTP) over Internet Protocol (TCP/IP).

34. A system as recited in claim 33, wherein the wireless communications protocol is selected from a group consisting of Wireless Application Protocol (WAP) and Handheld Device Transport Protocol (HDTP).

* * * * *

US005961593A

United States Patent [19]

Gabber et al.

[11] Patent Number: 5,961,593

[45] Date of Patent: *Oct. 5, 1999

[54] **SYSTEM AND METHOD FOR PROVIDING ANONYMOUS PERSONALIZED BROWSING BY A PROXY SYSTEM IN A NETWORK**

[75] Inventors: **Eran Gabber**, Summit; **Phillip P. Gibbons**, Westfield, both of N.J.; **Yossi Matias**, Potomac, Md.; **Alain J. Mayer**, New York, N.Y.

[73] Assignee: **Lucent Technologies, Inc.**, Murray Hill, N.J.

[*] Notice: This patent issued on a continued prosecution application filed under 37 CFR 1.53(d), and is subject to the twenty year patent term provisions of 35 U.S.C. 154(a)(2).

[21] Appl. No.: **08/787,557**

[22] Filed: **Jan. 22, 1997**

[51] **Int. Cl.**[6] **G06F 13/14**; H04L 9/32

[52] **U.S. Cl.** **709/219**; 709/225; 710/36; 370/85.13; 370/329; 380/25; 380/49

[58] **Field of Search** 380/25, 49; 370/329, 370/85.13; 710/1, 36; 709/225, 219; 395/200.16, 200.3, 200.57

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,550,984	8/1996	Gelb	395/200.17
5,673,322	9/1997	Pepe et al.	380/49
5,729,537	3/1998	Billstrom	370/329
5,742,762	4/1998	Scholl et al.	395/200.3
5,768,391	6/1998	Ichikawa	380/49

FOREIGN PATENT DOCUMENTS

PCT 97/15885	5/1997	WIPO	G06F 13/00

OTHER PUBLICATIONS

Article entitled "Security Without Identification: Transaction Systems to Make Big Brother Obsolete" by David Chaum, from Communications of the ACM, Oct. 1985, pp. 1030–1044.

Article entitled "Privacy–Enhancing Technologies for the Internet" by Ian Goldberg, David Wagner and Eric Brewer of the University of California, Berkeley.

Article entitled "NetBill Security and Transaction Protocol" by Benjamin Cox. J.D. Tygar and Marvin Sirbu of Carnagie Mellon University.

Article entitled "How to Construct Random Functions" from the Journal of the Association for Computing Machinery, by Oded Goldreich, Shafi Goldwasser and Silvio Micali, Oct. 1986, pp. 792–807.

Article entitled "Proceedings Symposium on Network and Distributed System Security" by the IEEE Computer Society Press: Feb. 22–23, 1996; pp. 1–16.

Article entitled "Anonymous Connections and Onion Routing" 1997 IEEE Symposium on Security and Privacy; by Paul F. Syverson, David M. Goldschlag and Michael G. Reed, Naval Research Library, pp. 1–11.

(List continued on next page.)

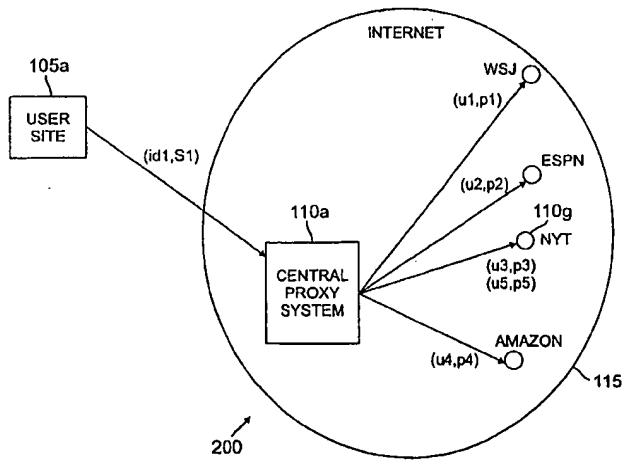
Primary Examiner—Thomas C. Lee

Assistant Examiner—Rehana Perveen

[57] **ABSTRACT**

For use with a network having server sites capable of being browsed by users based on identifiers received into the server sites and personal to the users, alternative proxy systems for providing substitute identifiers to the server sites that allow the users to browse the server sites anonymously via the proxy system. A central proxy system includes computer-executable routines that process site-specific substitute identifiers constructed from data specific to the users, that transmits the substitute identifiers to the server sites, that retransmits browsing commands received from the users to the server sites, and that removes portions of the browsing commands that would identify the users to the server sites. The foregoing functionality is performed consistently by the central proxy system during subsequent visits to a given server site as the same site specific substitute identifiers are reused. Consistent use of the site specific substitute identifiers enables the server site to recognize a returning user and, possibly, provide personalized service.

55 Claims, 6 Drawing Sheets



OTHER PUBLICATIONS

Article entitled "Frequently Asked Questions about Mixmaster Remailers" FAB Verson 1.8, Jul. 4, 1996 by Lance Cottrell.

"How to Make Personalized Web Browsing Simple, Secure, and Anonymous" by Eran Gabber, Phillip B. Gibbons, Yossi Matias, Alain Mayer; pp. 17–31; Feb. 1997.

"How to be Virtually Anonymous" by Randal L. Schwartz; Feb. 1997; pp. 30–33.

"Private Web Browsing " by Paul F. Syverson, Michael G. Reed, David M. Goldschlag; 1997; pp. 237–248.

"Anonymous Connections and Onion Routing " by Paul F. Syverson, David M. Goldschlag and Michael G. Reed; May 1997; pp. 44–54.

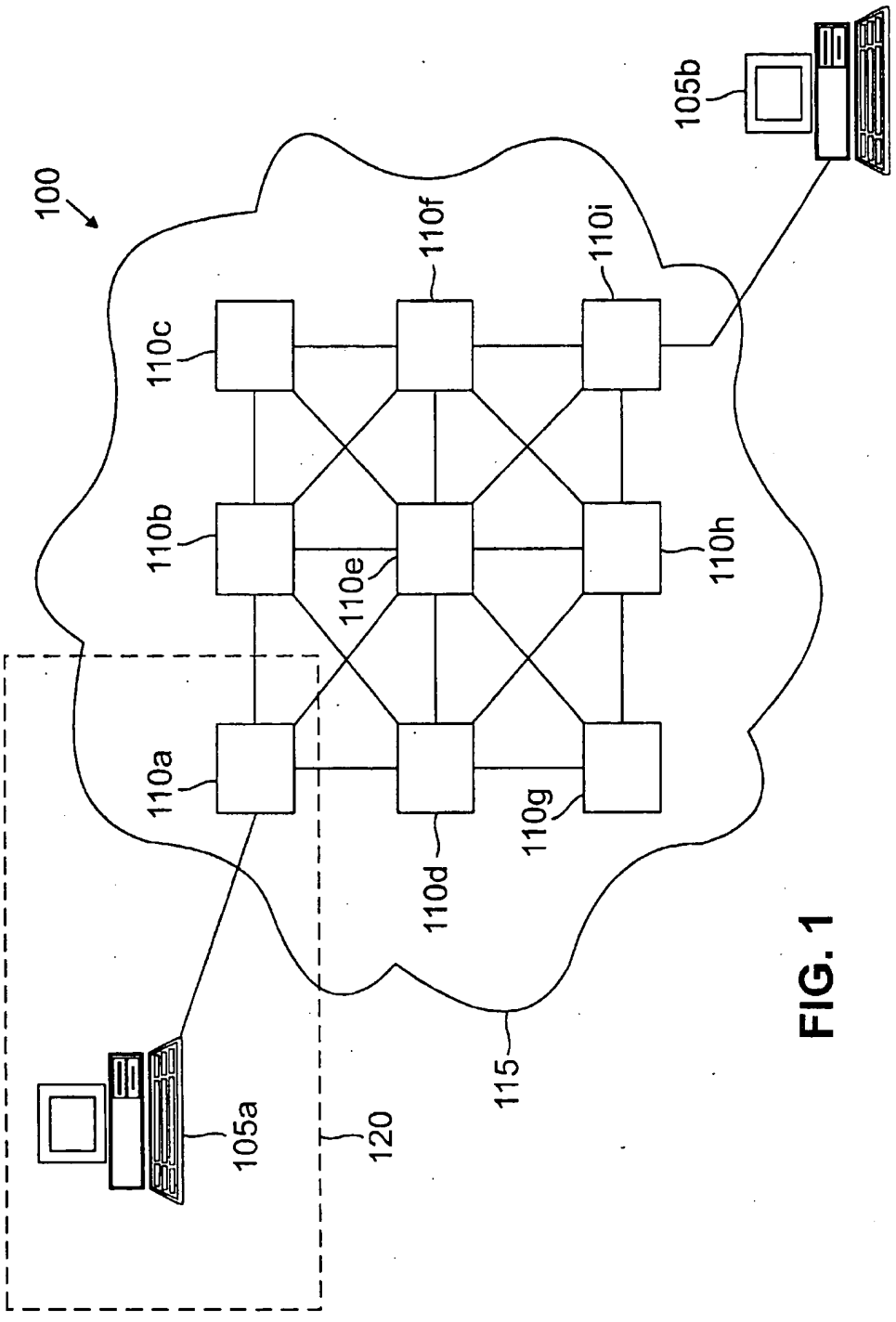

100
105b
110f
110i
110c
110b
110e
110h
110a
110d
110g
115
105a
120

FIG. 1

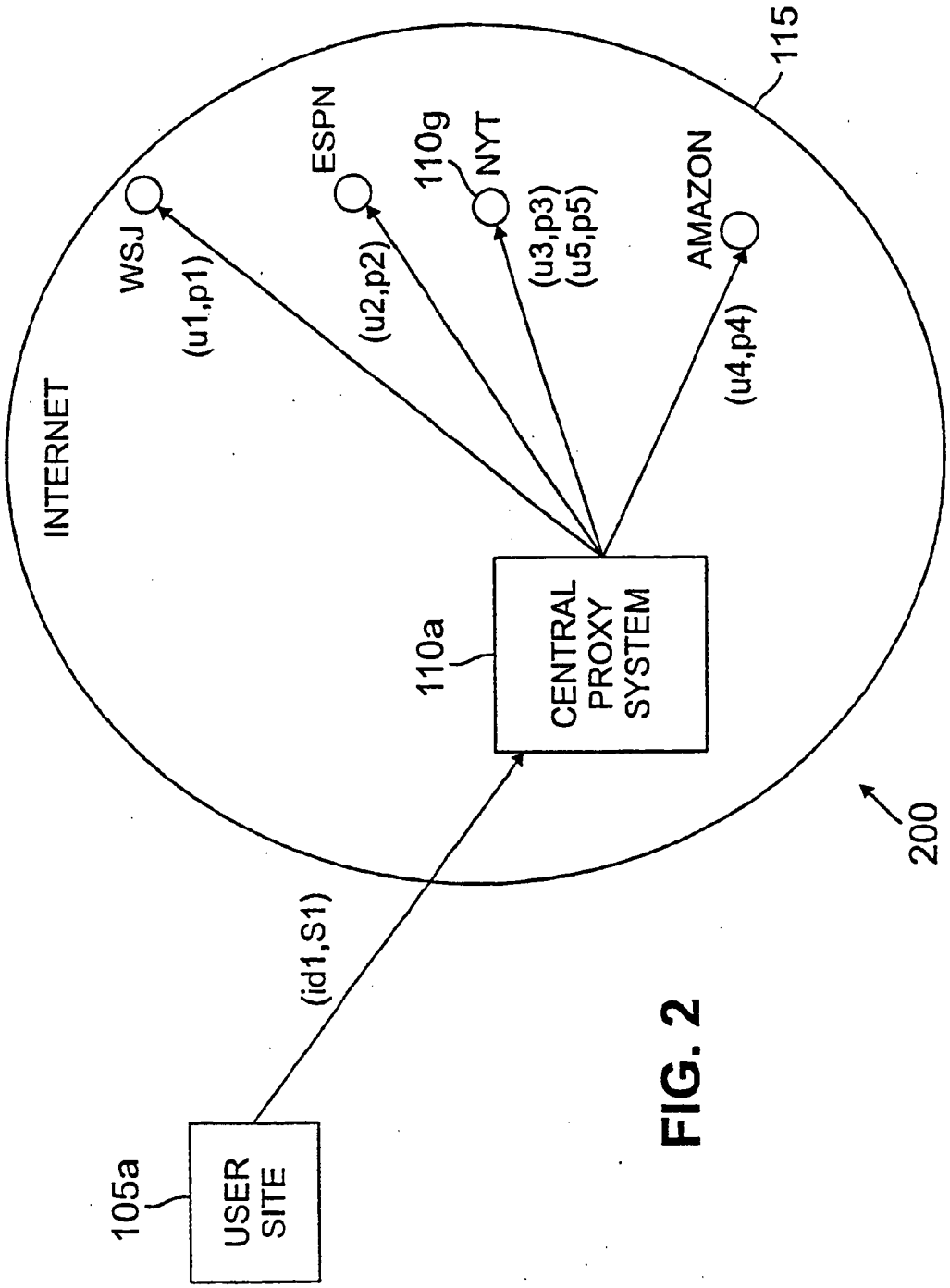

115
INTERNET
WSJ
(u1,p1)
ESPN
(u2,p2)
110g
NYT
(u3,p3)
(u5,p5)
AMAZON
(u4,p4)
110a
CENTRAL PROXY SYSTEM
200
(id1,S1)
105a
USER SITE

FIG. 2

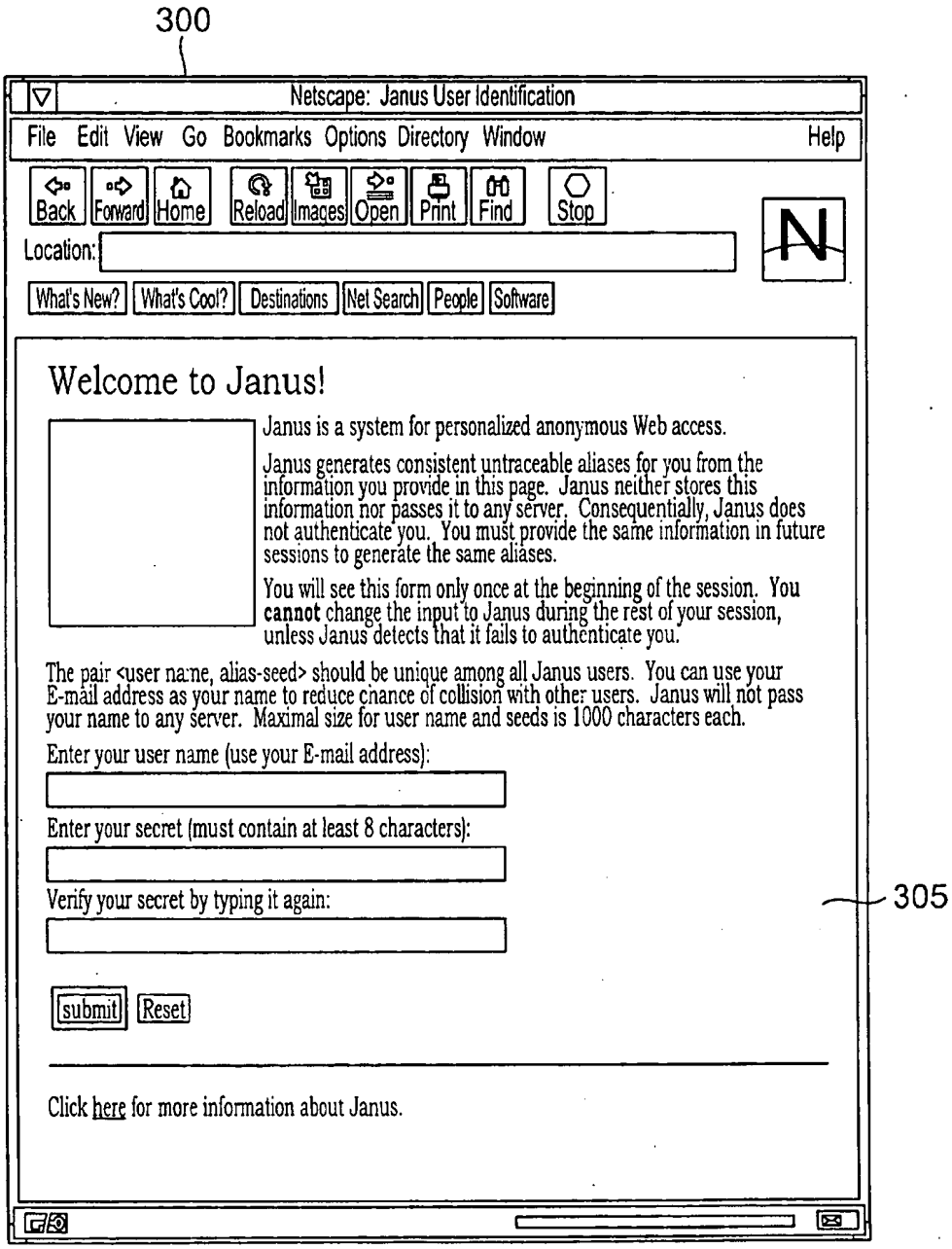

300
Netscape: Janus User Identification
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Reload Images Open Print Find Stop
Location:
What's New? What's Cool? Destinations Net Search People Software
Welcome to Janus!
Janus is a system for personalized anonymous Web access.
Janus generates consistent untraceable aliases for you from the information you provide in this page. Janus neither stores this information nor passes it to any server. Consequentially, Janus does not authenticate you. You must provide the same information in future sessions to generate the same aliases.
You will see this form only once at the beginning of the session. You cannot change the input to Janus during the rest of your session, unless Janus detects that it fails to authenticate you.
The pair <user name, alias-seed> should be unique among all Janus users. You can use your E-mail address as your name to reduce chance of collision with other users. Janus will not pass your name to any server. Maximal size for user name and seeds is 1000 characters each.
Enter your user name (use your E-mail address):
Enter your secret (must contain at least 8 characters):
Verify your secret by typing it again:
305
submit Reset
Click here for more information about Janus.

FIG. 3

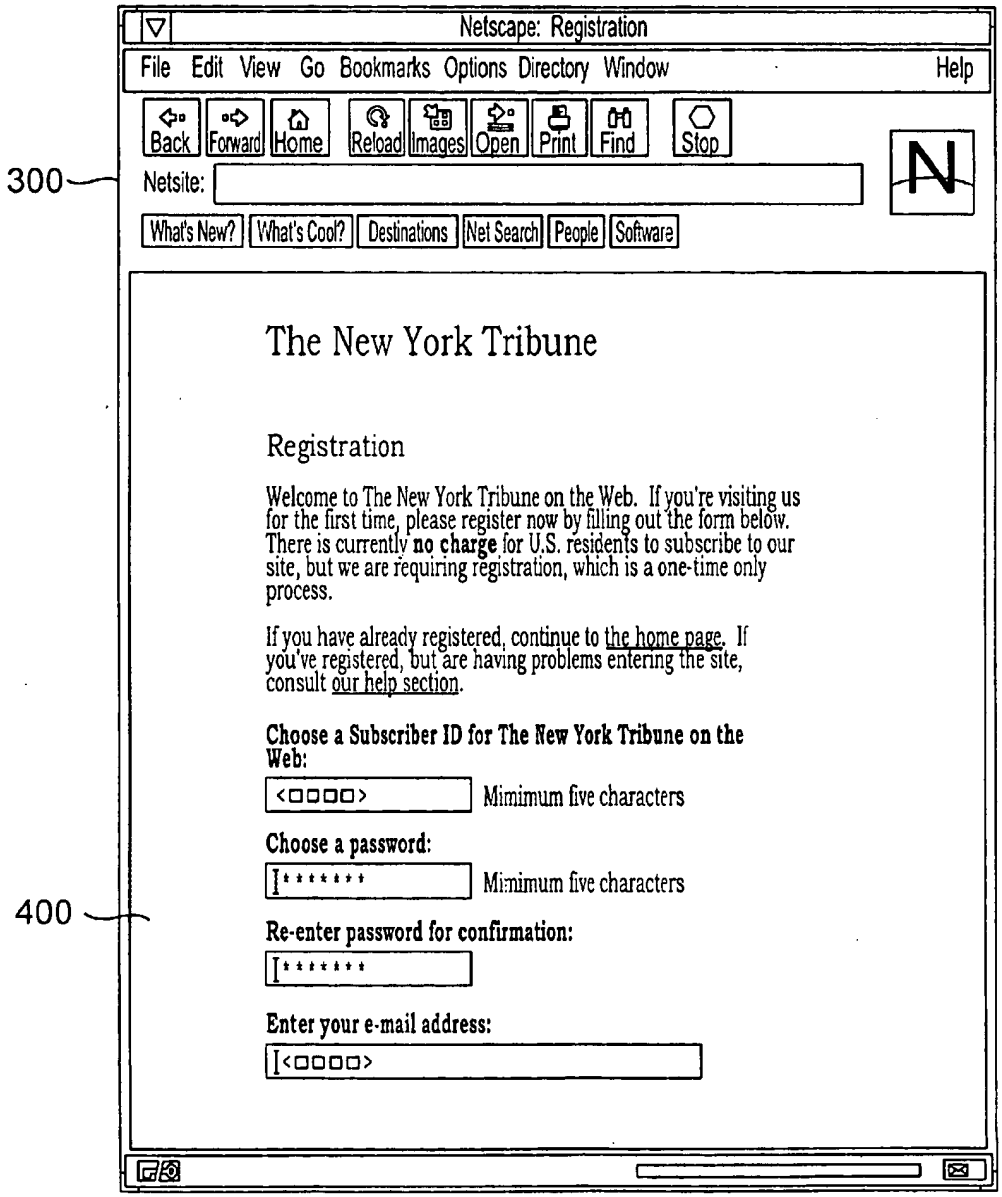

Netscape: Registration
File Edit View Go Bookmarks Options Directory Window Help
Back Forward Home Reload Images Open Print Find Stop
300
Netsite:
What's New? What's Cool? Destinations Net Search People Software
The New York Tribune
Registration
Welcome to The New York Tribune on the Web. If you're visiting us for the first time, please register now by filling out the form below. There is currently no charge for U.S. residents to subscribe to our site, but we are requiring registration, which is a one-time only process.
If you have already registered, continue to the home page. If you've registered, but are having problems entering the site, consult our help section.
Choose a Subscriber ID for The New York Tribune on the Web:
Mimimum five characters
Choose a password:
Minimum five characters
400
Re-enter password for confirmation:
Enter your e-mail address:

FIG. 4

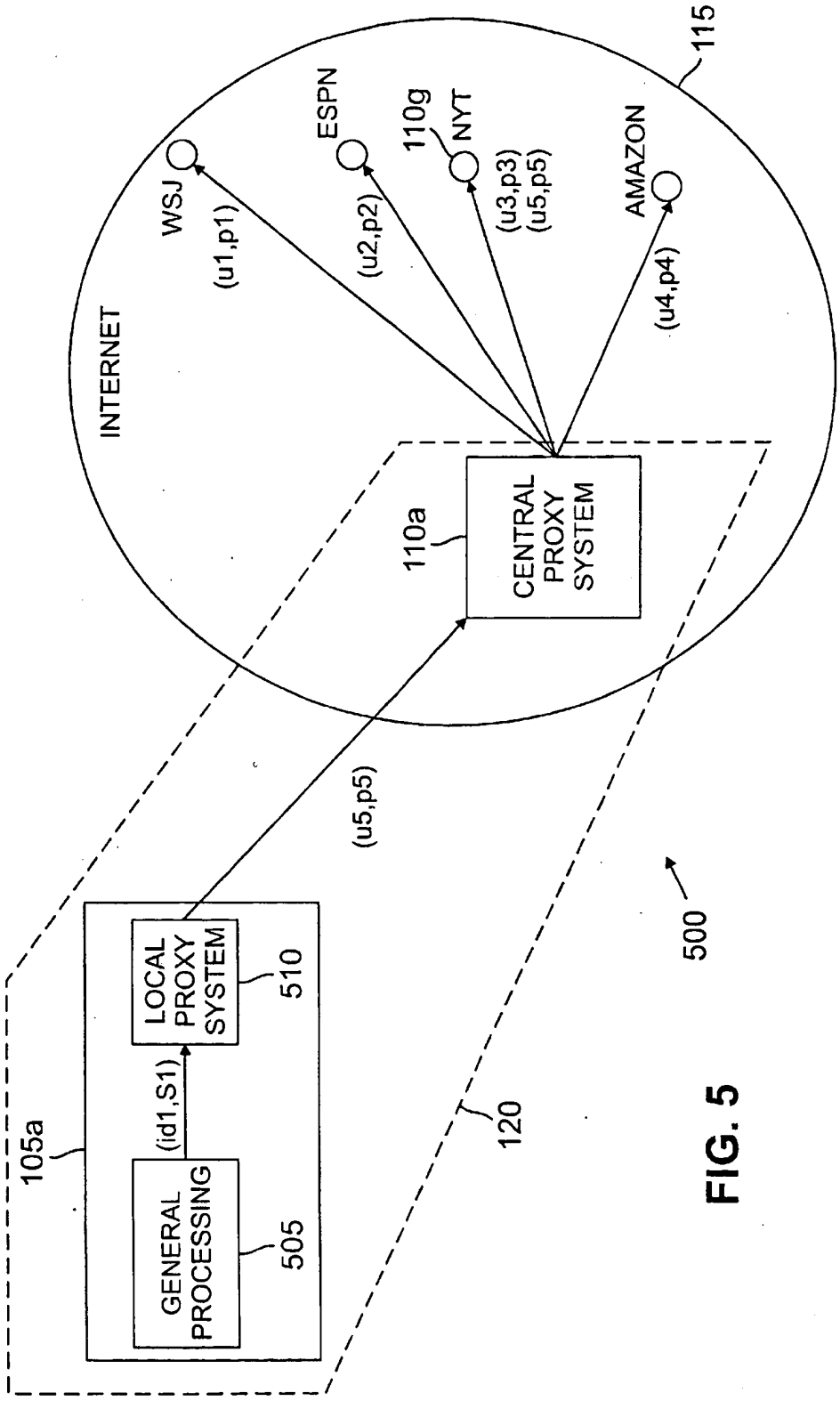

115
INTERNET
WSJ
(u1,p1)
ESPN
(u2,p2)
110g
NYT
(u3,p3)
(u5,p5)
AMAZON
(u4,p4)
110a
CENTRAL PROXY SYSTEM
(u5,p5)
500
120
105a
LOCAL PROXY SYSTEM
510
(id1,S1)
GENERAL PROCESSING
505

FIG. 5

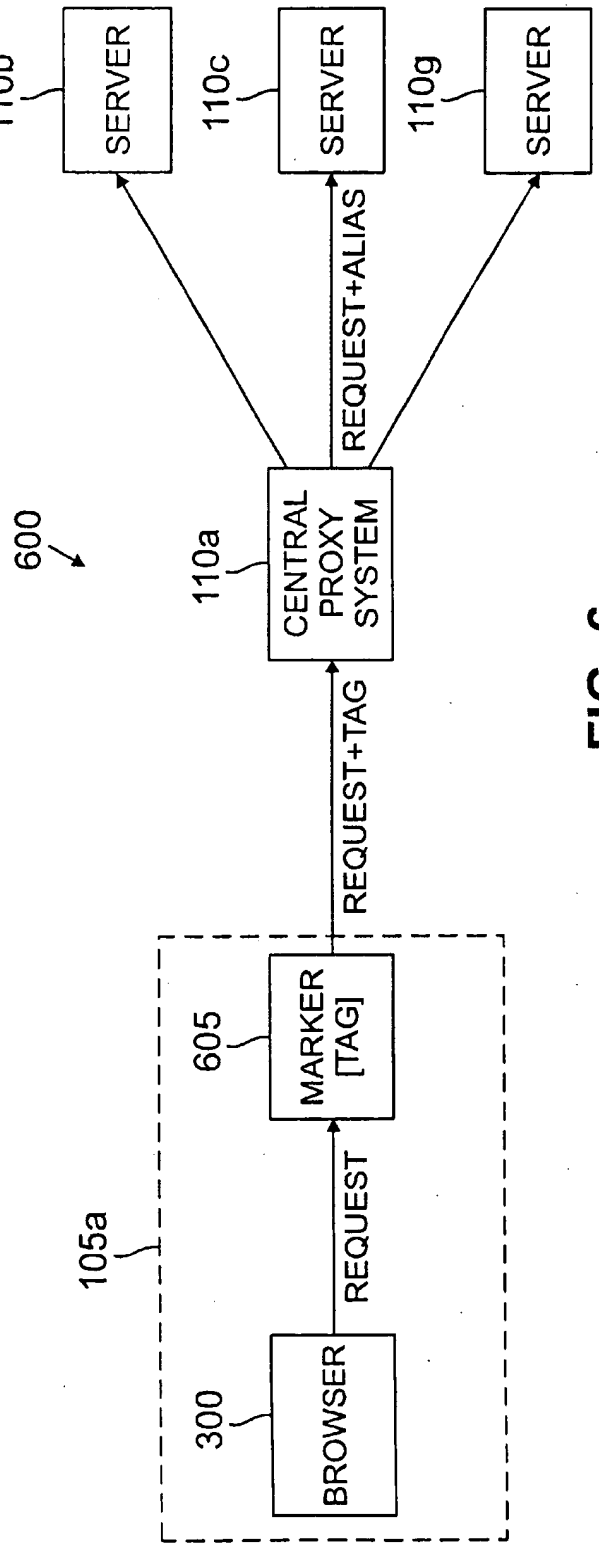

110b
SERVER
110c
SERVER
110g
SERVER
REQUEST+ALIAS
600
110a
CENTRAL PROXY SYSTEM
REQUEST+TAG
605
MARKER [TAG]
105a
REQUEST
300
BROWSER

FIG. 6

SYSTEM AND METHOD FOR PROVIDING ANONYMOUS PERSONALIZED BROWSING BY A PROXY SYSTEM IN A NETWORK

TECHNICAL FIELD OF THE INVENTION

The present invention is directed, in general, to networks and, more specifically, to a system and method that allows a user to browse personalized server resources on a network anonymously.

BACKGROUND OF THE INVENTION

The Internet is a well-known collection of networks (e.g., public and private data communication and multimedia networks) that work together (cooperate) using common protocols to form a world wide network of networks.

In recent years, the availability of more efficient, reliable and cost-effective computers and networking tools have allowed many companies and individuals (collectively, "users") to become involved in an ever growing electronic marketplace. The immeasurable gains in technology experienced by the computer industry overall have allowed these users to rely on commercially available computers, such as personal computers ("PCS"), to meet their information processing and communication needs. To that end, PC manufacturers equip most PCS with an interface that may be used for communication over networks, such as the Internet.

The Internet continues to increase its position as an integral place for businesses that offers information and services to potential customers. Popular examples of such businesses are news providers (e.g., www.cnn.com (the Cable News Network), www.nytimes.com (the New York Times), www.wsj.com (the Wall Street Journal), www.ft-.com (Financial Times Magazine), www.businessweek.com (Business Week Magazine)); car manufacturers (e.g., www.ford.com/us (the Ford Motor Company), www.gm-.com (the General Motor Company), www.toyota.com (the Toyota Motor Company)); book stores (e.g., www.amazon-.com (Amazon.com books)); software providers (e.g., www-.microsoft.com (the Microsoft software company)) and many more.

Most often, such a business sets up a home page on the World Wide Web (a "web-site," the World wide Web is a logical overlay of the Internet). The web-site constitutes an electronically-addressable location that may be used for promoting, advertising and conducting business. Potential electronic customers use web-browsers (e.g., NETSCAPE NAVIGATOR®, MICROSOFT EXPLORER®, etc.) to access the information offered on those web-sites.

An increasing number of web sites offer personalized services that may include "personalized web pages" customized to a user's interests, with hyper-links (a reference or link from some point in one hypertext document to some point in another document or another place in the same document—often displayed in some distinguishing way (e.g., in a different color, font or style)) and displayed messages tailored according to the user's preferences. Such preferences can be ascertained by having a user establish an account with that web-site. This allows the web-site to store information about the user's previous visits, either by tracking the hyper-links the user followed or through explicit dialogs with the user. For example, the Wall Street Journal provides a "personalized journal" to each user, where the sequence and selection of sections is customized. In order to open an account, the user typically has to complete a form electronically, providing a user name, a password, an electronic-mail ("e-mail") address, etc. The latter is often

used by the web-site to send back information not provided on the web-site itself to the user.

Given the inherent lack of privacy of electronic communication over the Internet generally, and, particularly, the World Wide Web, it has long been felt that a system that could ensure private electronic communication would be highly advantageous. As an example of the problem, consider the plight of a customer that would like to browse the World Wide Web in a safe and private (anonymous) manner, visiting sites that provide personalized service. The customer would like to establish accounts on web-sites without revealing his true identity, and without reusing the same user names, passwords, for multiple sites. Customers should refrain from reusing the same user names and passwords at multiple sites to avoid a security breach at one site to affect other sites; additionally refraining from using such user names and passwords limits the ability of multiple sites from colluding to combine customer information and build dossiers on particular customers.

Typically, the customer visits many of these web-sites, and inventing and remembering new user names and passwords for each web-site becomes tedious. Moreover, many of these web-sites require the customer to include his e-mail address with his user name and password—by providing his e-mail address, the customer reveals his identity.

In addition, there are commercial products available that allow web-sites to track their clients and visitors. Such tracking can be made even when no voluntary information is provided by the user and no form is filled out. Examples of such systems are "Webreporter," which is available from OPENMARKET, INC., and "SiteTrack," which is available from GROUP CORTEX, whose advertisement reads as follows:

> "Identify who is visiting your site. Record the actual number of people that visit. Find which links they follow and trace their complete path. Learn which site users came from and which site they depart to . . . "

These products are made possible because the hypertext transport protocol ("HTTP-protocol"), on which the World Wide Web is largely based, allows specific information to flow back from the user to the web-site. This can include for example, the user's e-mail address, the last web-site he came from, and information about the user's software and host-computer. Other pertinent user information may be sent by the web-site to the user browser using what are commonly referred to as "cookies" (pieces of information that web-sites may store at the user's browser). On subsequent visits to the web-site, the user's browser sends back information to the web-site without the user's knowledge.

From the foregoing, it is apparent that what is needed in the art is a scheme that provides anonymous personalized web browsing that satisfies two seemingly conflicting objectives, namely, providing user privacy and user identification.

SUMMARY OF THE INVENTION

To address the above-discussed deficiencies of the prior art, the present invention introduces a proxy system that performs two basic functions: (1) automatic substitution of user-specific identifiers such that server sites (e.g., web sites, junction points, intelligent portal devices, routers, network servers, etc.) within a network are prevented from determining the true identity of the user browsing (accessing, locating, retrieving, reading, contacting, etc.) the sites; and (2) automatic stripping of any other information associated with browsing commands that would allow the server sites

to determine the true identity of the user browsing the server sites. An important aspect of the present invention is that the foregoing functions are performed consistently by the proxy system during subsequent visits to the server site (the same substitute identifiers are used on repeat visits to the server site; the server site also cannot distinguish between information supplied by the user and the proxy system, thus the proxy system is transparent to the server site). The present invention therefore not only introduces anonymous browsing, but also personalization based upon the consistent use of substitute identifiers.

It should be noted that the term "true," as used herein, means accurate, actual, authentic, at least partially correct, genuine, real or the like, the term "or," as used herein, is inclusive, meaning and/or; and the phrase "associated with" and derivatives thereof, as used herein, may mean to include within, interconnect with, contain, be contained within, connect to or with, couple to or with, be communicable with, juxtapose, cooperate with, interleave, be a property of, be bound to or with, have, have a property of, or the like.

As is described in greater detail hereinbelow, the principles of the present invention address the conflicting objectives of user privacy and user identification described hereinabove by providing a proxy system, a peripheral proxy system, and a method of providing substitute identifiers to a server site that allow users to browse the same anonymously via the proxy system.

In one embodiment, the present invention provides, for use with a network having server sites capable of being browsed by users based on identifiers received into the server sites and personal to the users, a central proxy system for providing substitute identifiers to the server sites that allow the users to browse the server sites anonymously via the central proxy system. According to various embodiments of the present invention, the substitute identifiers may be suitably constructed by the user site or a routine associated with the central site (advantageous ways (functions) of constructing the substitute identifiers are described hereinafter). The exemplary central proxy system includes: (1) a computer-executable first routine that processes (receives, accepts, obtains, constructs, produces, etc.) site-specific substitute identifiers constructed from data specific to the users, (2) a computer-executable second routine that transmits substitute identifiers to the server sites and thereafter retransmits browsing commands received from the users to the server sites and (3) a computer-executable third routine that removes (and possibly substitutes) portions of the browsing commands that would identify the users to the server sites. "Include" and derivatives thereof, as used herein, means inclusion without limitation.

In one embodiment, the first of the two above-enumerated basic functions is performed external to the central proxy system, while in another it is performed, at least in part, within the central proxy system. The central proxy system processes and forwards the substitute identifiers as appropriate and directly performs the second of the above-enumerated basic functions by stripping other information that would tend to identify the users. An Internet Access Provider ("ISP"), such as NETCOM®, or a networking service, such as AMERICA ONLINE® or COMPUSERVE® can advantageously employ the central proxy system to provide anonymous retransmission of browsing commands by their users.

It is important to understand that subsequent use of the proxy system by a "same" user to a "same" server site will cause the proxy system to construct (directly or indirectly) and use the same (site-specific) substitute identifiers. Typically, the proxy system functions as a conduit communicating messages between the user and the server. Depending upon the embodiment, the proxy system may remove or substitute some portion of messages communicated by the user to the server to ensure anonymity.

An alternative advantageous embodiment of the present invention may be provided in the form of a peripheral proxy system designed for use with a network having a server site capable of being browsed by users based on identifiers received into the server site and personal to the users. The peripheral proxy system includes: (1) a computer-executable first routine that constructs a particular substitute identifier from data received from a particular user and (2) a computer-executable second routine that transmits the particular substitute identifier to the central proxy system, the central proxy system retransmitting the particular substitute identifier to the server site and thereafter retransmitting browsing commands received from the particular user to the server site. According to this embodiment, the first routine may be associated, at least in part, with the user site, which distributes the basic functions of the present invention over multiple computer systems.

The foregoing has outlined, rather broadly, preferred and alternative features of the present invention so that those skilled in the art may better understand the detailed description of the invention that follows. Additional features of the invention will be described hereinafter that form the subject of the claims of the invention. Those skilled in the art should appreciate that they can readily use the disclosed conception and specific embodiment as a basis for designing or modifying other structures for carrying out the same purposes of the present invention. Those skilled in the art should also realize that such equivalent constructions do not depart from the spirit and scope of the invention in its broadest form.

BRIEF DESCRIPTION OF THE DRAWINGS

For a more complete understanding of the present invention, reference is now made to the following descriptions taken in conjunction with the accompanying drawings, wherein like numbers designate like objects, and in which:

FIG. 1 illustrates a high-level block diagram of an exemplary distributed network with which the principles of the present invention may be suitably used to provide either a central or a peripheral proxy system for allowing users to provide substitute identifiers to server sites of a network to browse anonymously;

FIG. 2 illustrates a block diagram of an exemplary sub-network of the distributed network of FIG. 1 showing a central proxy system that includes each of a user site, a central proxy system and a plurality of illustrative server sites according to the principles of the present invention;

FIG. 3 illustrates an exemplary full screen window of a proxy system according to the principles of the present invention;

FIG. 4 illustrates an exemplary full screen window of an interface of a particular server site according to the principles of the present invention;

FIG. 5 illustrates a block diagram of an exemplary sub-network of the distributed network of FIG. 1 showing a peripheral proxy system that includes each of a user site, a central proxy system and a plurality of illustrative server site according to the principles of the present invention; and

FIG. 6 illustrates a block diagram of an exemplary sub-network of the distributed network of FIG. 1 including each

of a user site, a central proxy system and a plurality of illustrative server sites according to an exemplary marker proxy embodiment of the present invention.

DETAILED DESCRIPTION

Referring initially to FIG. **1**, illustrated is a high-level block diagram of an exemplary distributed network (generally designated **100**) with which the principles of the present invention may be suitably used to provide either a central or a peripheral proxy system. Distributed network **100** illustratively includes a plurality of computer sites **105** to **110** that are illustratively associated by Internet **115**. Internet **115** includes the World Wide Web, which is not a network itself, but rather an "abstraction" maintained on top of Internet **115** by a combination of browsers, server sites, HTML pages and the like.

According to the illustrated embodiment, either proxy system provides substitute identifiers to one or more of a plurality of server sites **110** of network **100**. The substitute identifiers allow user sites (and, hence, users (not shown)) to browse the server sites anonymously via the proxy system. Consistent use of the same (site-specific) substitute identifiers at a particular server site personalizes browsing. For purposes of illustration, site **105***a* is assumed throughout this document to be a user site, site **110***a* is assumed to be a central proxy site, and site **110***g* is assumed to be a server site.

Those of skill in the pertinent art will understand that FIG. **1** is illustrative only, in other configurations, any of sites **105** to **110** may be a user, a central proxy or a server site, or a combination of at least two of the same. "Server site," as the term is used herein, is construed broadly, and may include any site capable of being browsed.

Although the illustrated embodiment is suitably implemented for and used over Internet **115**, the principles and broad scope of the present invention may be associated with any appropriately arranged computer, communications, multimedia or other network, whether wired or wireless, that has server sites capable of being browsed by users based on identifiers received into the server sites and that are personal to the users. Further, though the principles of the present invention are illustrated using a single user site **105***a*, a single central proxy site **110***a* and a single server site **110***g*, alternate embodiments within the scope of the same may include a plurality of user, central proxy or server sites.

Exemplary network **100** is assumed to include a plurality of insecure communication channels that operate to intercouple ones of the various sites **105** to **110** of network **100**. The concept of communication channels is known and allows insecure communication of information among ones of the intercoupled sites (the Internet employs conventional communication protocols that are also known). A distributed network operating system executes on at least some of sites **105**, **110** and may manage the insecure communication of information therebetween. Distributed network operating systems are also known.

According to exemplary central proxy system **110***a* of the present invention, which is discussed in detail with reference to FIG. **2**, substitute identifiers may be suitably indirectly provided by central proxy system **110***a* to server site **110***g* (recall that substitute identifiers allow user site **105***a* to browse server site **110***g* anonymously). One or more site-specific substitute identifiers are suitably provided or constructed from data specific to user **105***a* either by user **105***a* or central proxy system **110***a*. Central proxy system **110***a* includes a plurality of executable routines—a first routine processes site-specific substitute identifiers constructed from data specific to user **105***a* (site-specific substitute identifiers may be suitably constructed by a central proxy site **110***a*, such as by a routine associated with central proxy system **110***a*); a second routine transmits the substitute identifiers to server site **110***g* (possibly via a plurality of intermediate user and server sites **105**, **110**) and thereafter retransmits browsing commands received from user site **105***a* to server site **110***g*; and a third routine removes (and possibly substitutes) portions of the browsing commands that would identify user site **105***a* to server site **110***g* (and the plurality of intermediate user and server sites **105**, **110**). The term "routine," as used herein, is construed broadly to not only include conventional meanings such as program, procedure, object, task, subroutine, function, algorithm, instruction set and the like, but also sequences of instructions, as well as functionally equivalent firmware and hardware implementations.

Alternatively, according to an exemplary peripheral proxy system (generally designated **120**) of the present invention, which is discussed in detail with reference to FIG. **5**, that is designed for use with network **100** again having a server site **110***g* capable of being browsed by a user site **105***a* based on substitute identifiers received into server site llog and that are personal to user site **105***a*. Exemplary peripheral proxy system **120** includes first and second executable routines. The first routine, which may advantageously reside in user site **105***a* or, alternatively, in central proxy system **110***a*, constructs a particular substitute identifier from data particular to user site **105***a*. The second routine, which may also advantageously reside in user site **105***a* or, partially, in user site **105***a* and central proxy system **110***a*, transmits the particular substitute identifier to central proxy system **110***a*. Central proxy system **110***a* then retransmits the particular substitute identifier to server site **110***g* and thereafter communicates (e.g., transmits, receives, etc.) information (e.g., browsing commands, data, etc.) between user site **105***a* to server site **110***g*.

According to the illustrated embodiment, peripheral proxy system **120** differs from central proxy system **110***a* by the location of execution of the first and second routines. In the illustrated central proxy embodiment, all routines are executed by central proxy system **110***a*, which means that all users must send user specific information to central proxy system **110***a*. In the illustrated peripheral proxy system **120**, the first and second routines may be executed in a proxy subsystem associated with user site **105***a*. In one advantageous embodiment, user system **105***a*'s user specific information (e.g., user identification, passwords, e-mail addresses, telephone numbers, credit card numbers, postal address, etc.) remain local, which will typically be more secure than central proxy system **110***a*.

As set forth hereinabove, an ISP, such as NETCOM®, or a networking service, such as AMERICA ONLINE® or COMPUSERVE®, can advantageously employ either exemplary proxy system (central or peripheral) to provide anonymous communication (transmission, reception, retransmission, etc) of browsing (e.g., accessing, selection, reading, etc.) commands between user sites and server sites.

An important aspect of the above-identified embodiments is the use of site-specific substitute identifiers to eliminate the need for a user to have to "invent" a new user name and password for each server site which requires the establishment of an account (e.g., the NEW YORK TIMES, the WALL STREET JOURNAL, the NEWSPAGE® and ESPN® sites). The illustrated embodiment generates secure substitute identifiers (e.g., alias user names, passwords, e-mail addresses, postal addresses, credit card numbers, etc.)

that are distinct and secure for the user. The user provides one or more character strings (which may be random) once, which may advantageously be at the beginning of a proxy system session. The proxy system uses the same to generate one or more secure site-specific substitute identifiers for the user—thereby freeing the user from the burden of inventing new and unique identifiers for each server site. Moreover, the user no longer has to type such secure identifiers every time the user returns to a particular server site requiring an account; instead the proxy system provides the appropriate secure identifiers automatically. In an advantageous embodiment to be described, the proxy system filters other identifying information (e.g., HTTP headers, etc.) sent by user site **105***a* while browsing server sites. It is important to keep in mind that server sites cannot typically distinguish between information supplied by proxy system **110***a* and information supplied by user site **105***a*—central proxy system **110***a* being transparent to server sites.

In one embodiment, the substitute identifiers are transmitted on demand from servers, without any intervention from the user. This process automates the response to a "basic authentication request," which is a common procedure used by servers to identify users on the World Wide Web. In this way, the user is not burdened by this activity.

According to the illustrated embodiment, to produce substitute identifiers the proxy system may suitably maintain secret information (secret to at least one server-site) in the form of user definable character strings. These character strings may be user defined and may be maintained in some conventional manner, such as storing the same to memory associated with the proxy system, or, advantageously, a function (described hereinafter) may be used to produce the substitute identifiers, at least in part, in association with the secret information. According to one approach, the proxy system maintains a conventional data structure to maintain the same, such as a database, data repository, an array, etc., or even an alias table, that may be used to map user information to their substitute, or alias, identifiers.

According to one advantageous embodiment, the user delivers its own secret (user definable character string) at the beginning of each session, which is used by the proxy system to generate, directly or indirectly, the substitute identifiers for the session. This option has the advantage that a user has the flexibility to choose different proxies at different times and there is no permanent secret information stored on the proxy system. In another related embodiment, the data comprises at least two secret user definable character strings, wherein the first routine processes substitute identifiers constructed in part from the at least two secret user definable character strings. Of course, alternate suitable approaches may be used to accomplish the purpose of providing anonymous personalized web browsing according to the present invention.

Turning now to FIG. **2**, illustrated is a block diagram of an exemplary sub-network (generally designated **200**) of distributed network **100**, wherein sub-network **200** includes user site **105***a*, central proxy system **110***a* and server site **110***g* (shown among a plurality of other illustrative server sites **110** of Internet **115**) according to the principles of the present invention.

For purposes of illustration, assume that user site **105***a* issues a command to access server site **110***g* (the NEW YORK TRIBUE web-site ("NYT")). Such access would be via central proxy system (server site) **110***a*, which ensures that user specific data concerning user site **105***a* is not communicated over the remainder of Internet **115**—there may be HTTP header fields, for example, that include data about user site **105***a* that central proxy system **110***a* filters.

Exemplary central proxy system **110***a* advantageously executes on a server site that is not associable with user site **105***a* by other sites over Internet **115**. According to an advantageous embodiment, central proxy system. **110***a* may be suitably distant, both physically and logically, from user site **105***a*—user site **105***a* does not access server-sites directly because the server-sites can determine both physically and logically the Internet Protocol ("IP")—address of the machine that made the request.

According to the exemplary embodiment, if user site **105***a*'s command to access NYT llog is user site **105***a*'s first request of the current session, central proxy system **110***a* will recognize the same, and display its own HTML-document, possibly on user site **105***a*'s browser.

Turning momentarily to FIG. **3**, illustrated is an exemplary full screen window of a conventional browser **300** ("NETSCAPE®") displaying an inlaid interface **305** ("JANUS℠") of central proxy system **110***a* according to the principles of the present invention. Exemplary interface **305** prompts a user of site **105***a* to enter user definable character strings, which according to the illustrated embodiment includes identification ("ID") data and secret ("S") data supplied by the user. Each user initially supplies a user ID (e.g., e-mail address) and a user S to allow one or more substitute identifiers to be chosen or constructed (site-specific substitute identifiers are suitably constructed from data specific to user **105***a* and a particular server site which user **105***a* intends to browse). Alternatively, other or further data supplied by the user may be appropriate in some applications (e.g., credit card number, post office address, handle, etc.).

According to the advantageous embodiment, substitute identifiers may be constructed (generated) using a suitable function that includes the features of anonymity, consistency, collision resistance and uniqueness, protection from creation of dossiers, and single secret and acceptability. Concerning anonymity, the identity of the user should be kept secret; that is, a server site, or a coalition of sites, cannot determine the true identity of the user from its substitute identification. Concerning consistency, for each server-site, each user should be provided with some substitute identifiers allowing the server site to recognize the user given the same, thereby enabling the server site to personalize the user's access and the user can thus be "registered" at the server site.

With respect to collision resistance and uniqueness, given a user's identity and a server site, a third party should not find a different user identity which results in the same alias (impersonation) for that server site. As to protection from creation of dossiers, the user is likely to be assigned a distinct alias (substitute identifier) for distinct server sites, so that a coalition of sites is unable to learn a user's habits and build a user profile (dossier) based on the set of sites accessed by the user. Lastly, single secret (user definable character string) and acceptability provides, given the user's identity and a single secret, automatic generation of secure, distinct aliases (substitute identifier) as needed for each server-site, transparent to the user—from the user's perspective, the user definable character string is equivalent to a universal password for a collection of server-sites.

According to this embodiment, a user ID is "corrupt" (not secret) if an adversary (one or more server sites desirous of identifying the user), E, has been able to read the user's secret, S. Alternatively, a user ID is "partially opened" (not fully secure) with respect to a particular server site, w, if E

has been able to read the alias password; a user ID is "opened" (not secure) with respect to w, if it is partially opened and E has been able to relate the alias password together with the alias user name to the user ID. Assuming that the function, T(), is defined as follows, T(user ID, web-site ("w"), S)=(substitute username, passwords), hence, T(id,w,S)=(Uw,Pw); and Tu(id,w,S)=Uw and Tp(id,w,S)=Pw.

Tu(id,w,S)=Uw=h(enc(k,id,f(s_1, w))) and

Tp(id,w,S)=Pw=h(enc(k,id,f(s_2,w))), wherein

id denotes user site **105***a*'s ID (e.g., e-mail address);

w denotes server site **110***g*'s domain name;

|| denotes the logical function of concatenation;

S denotes k||s_1||s_2 a user site **105***a* definable character string;

xor denotes the Boolean function of exclusive or;

f(k,x) denotes a suitably arranged function for generating pseudo-random values, and may be selected from a group of functions, such as des(k,h(x),x);

enc(k,x,r) denotes r||(f(k,r)xor x);

h() denotes a collision-resistant hash function, such as MD5; and

des(k,i,x) denotes DES encryption in cipher block chaining ("CBC") mode, which are known, of information x using key k and an initialization vector i.

Both Tu() and Tp() may suitably truncate the result of the hashing function, h(), to fit the longest allowed user name or password for the particular server site.

Relating this function, T(), to the above-identified and described features yields the following:

1. E can only guess at the identity, ID, of a user which is only partially opened and uncorrupted.

2. T() is a deterministic function and E can only guess at the alias-password of a user which is unopened and uncorrupted.

3. Given w and an uncorrupted and unopened user ID, E can only guess at the ID and S.

4. For an uncorrupted user ID and w, T(id,w,S) does not give to E information about T(id,w',S) for any w' not equal to W.

5. The range of T(id,w,S) is such that it is accepted by server sites as a valid username and password—implying a limited length string of printable characters.

Those skilled in the pertinent art will understand that alternate suitable functions may replace or be used in association with the foregoing according to the principles of the present invention.

Use of the foregoing exemplary substitute identifier constructing function, and for that matter, any other suitably arranged function for constructing substitute identifiers according to the present invention, operates to foster the above-identified features of anonymyzed and personalized browsing. The present invention provides the ability to anonymously visit a server site a first time via site-specific substitute identifiers, to interact with the server site as a function thereof, and to re-visit the server site on subsequent occasions using the same site-specific substitute identifiers, interacting with the server site as a return customer—possibly receiving personalized attention—as a function of the recognized substitute identifiers. Simply stated, the substitute identifiers are constructed consistently, and in advantageous embodiments in a site-specific manner.

In one embodiment of the present invention, the substitute identifiers include site-specific substitute user names and site-specific substitute user passwords. "Site-specific" means that the names and passwords vary from site to site, depending perhaps upon the address of each site. This may complicate the task of creating a dossier relative to a given user. In a related embodiment, the first routine constructs site-specific substitute e-mail addresses for user site **105***a* from the site-specific data. In an alternate advantageous embodiment, the first routine constructs the site-specific substitute identifiers from addresses of the server sites—of course, site-specific information other than the address of the site may be used to construct the substitute identifiers.

If this is the first contact of the user with central proxy system **110***a*, then the user may suitably generate a user defined character string (secret) at random and store the same locally. In one advantageous embodiment, the first routine processes substitute identifiers that may be constructed by applying pseudo-random and hash functions (e.g., T() function set forth hereinabove) to the data received from user site **105***a*—those skilled in the art are familiar with the structure and operation of pseudo-random and hash functions and their utility. The important aspect of this and related embodiments is that the present invention is adapted to take advantage of current and later-discovered functions to enhance anonymity and security.

Alternatively, if this is the first contact of a current session then the user may suitably enclose the stored user defined character string to central proxy system **110***a*. Nonetheless, browser **300** sends interface **305** together with a user's ID and other user definable character string to central proxy system **110***a*. Central proxy system **110***a* receives this information and may use the same for the rest of the session.

In one advantageous embodiment, the first routine receives or generates session tags that are added to the browsing commands, central proxy site **110***a* employing the session tags to associate the substitute identifiers with each of the browsing commands—the session tags, while not necessary to the present invention, provide one manner that allows user sites **105***a* to supply their data only once, usually at the beginning of each session. In a related advantageous embodiment, central proxy site **110***a* includes a data store that is capable of containing session information specific to user sites **105***a* and accessible by server sites **110***g*.

In one advantageous embodiment, the second routine described above, which may be local to the central proxy system **110***a*, transmits the substitute identifiers to server site **110***g*. In a further advantageous embodiment, the second routine transmits the substitute identifiers to server site **110***g* based on alphanumeric codes supplied in fields of web-pages **305** by the users. The alphanumeric codes prompt the second routine as to how and where to locate the substitute identifiers, removing the users from actually having to provide the substitute identifiers directly. Of course, the alphanumeric codes may be supplied in a different form. In a related, more specific embodiment, the users manually place the alphanumeric codes in the fields of web-pages **305**. Of course, the present invention encompasses intelligent parsing of the fields of web pages **305** to determine automatically how and where the alphanumeric codes should be located. Those skilled in the art are familiar with the Internet in general, the World Wide Web in particular and the way in which the structure of the World Wide Web promotes "browsing." The present invention finds apparent utility in conjunction with the Internet and the World Wide Web, however, those skilled in the art will readily understand that the present invention has advantageous application outside of the Internet as well in any suitably arranged computer, communications, multimedia or like network configuration.

Nonetheless, after central proxy system **110***a* obtains the required information about the user, the above-described

third routine removes portions of the browsing commands that would identify user site **105***a* to server site **110***g*, and forwards user site **105***a*'s original request for access to NYT-site **110***g* (e.g., using an HTTP get-request)—thereby selectively excluding from the request header-fields or the like that may identify the user.

If this is the user's first visit to NYT-site **110***g*, then it may suitably provide the user with an electronic form prompting, for example, for a user name, a password and an e-mail address in order to establish an account. Turning momentarily to FIG. **4**, illustrated is exemplary full screen window of conventional NETSCAPE® browser **300** displaying an inlaid interface **400** ("THE NEW YORK TRIBUNE") of server site **110***g* according to the principles of the present invention.

Now, instead of having to provide a unique user name and a secret password, the user may suitably provide these fields with simple escape strings (e.g., "<uuuu>" and "<pppp>"). More specifically, the alphanumeric codes above-described may be suitably arranged into such escape sequences—those skilled in the art are familiar with escape sequences. These strings are recognized by central proxy site **110**a which uses user site **105***a*'s user name and secret (user definable character string) along with the domain-name of the NEW YORK TRIBUNE and computes substitute identifiers (e.g., alias user name, **u3**, and alias password, **p3**, in FIG. **2**, etc.), such as by function T(ID, secret, domain-name). The site-specific substitute identifiers may be sent to a particular server site by central proxy system **110***a* using the same mechanism that the user would submit input to the particular server site. In other words, proxy system **110***a* receives information communications, such as browsing commands, from user site **105***a* intended for server site **110***g*, and retransmits the same to server site **110***g*—central proxy system **110***a* functioning as a transparent conduit for anonymizing and, through consistent generation of site-specific substitute identifiers, personalizing server site browsing.

On a subsequent visit to NYT-site **110***g*, which will require that user site **105***a* authenticate itself (response to the first get-request forwarded to NYT-site **110***g* by central proxy system **110***a*), central proxy system **110***a* may be suitably operative to automatically recompute **u3** and **p3** and reply by sending these values back to NYT-site **110***g* (re-sending the get-request). User site **105***a* is thereby freed from the burden of remembering the user name and password of its NYT-site **110***g* account. To summarize, the protocol, which may be suitably executed without involving user site **105***a*, includes: (1) a step of NYT-site server **110***g* requesting an authentication from central proxy site **110***a* by failing the first get request; (2) central proxy site **110***a* recomputing the substitute identifiers (e.g., (alias-user name, alias-password)=T(ID, secret, domain-name), or the like); (3) central proxy site **110***a* replying by re-sending the get with the same substitute identifiers.

The substitute identifiers are consistent in the sense that the substitute identifiers are presented on subsequent visits to the same server site by user **105***a*. Consistent substitute identifiers allow server sites to recognize returning users and provide personalized service to them. In one embodiment, the second routine transmits the substitute identifiers on demand from servers, without any intervention from user **105***a*. This process automates the response to a "basic authentication request," which is a common procedure used by servers to identify users **105***a* on the World Wide Web. In this way, user **105***a* is not burdened by this activity. In this embodiment, the second routine may have to re-transmit the original user request along with the substitute identifier to the server.

It should be noted that many servers require a valid e-mail address for creating an account—users cannot use their true e-mail address for this purpose since it uniquely identifies them. The proxy system of the present invention may suitably solve this problem by creating an alias e-mail address for user site **105***a* and store e-mail in an electronic mailbox. In one advantageous embodiment central proxy system **110***a* includes a data store capable of containing e-mail destined for the users, thereby preventing server sites from contacting users directly. Contrary to prior art anonymous re-mailers, the present embodiment is not required to rely on having to store any translation tables (which may be large and vulnerable) from alias to true user identifiers in central proxy system **110***a*. This embodiment is inherently securer than prior art approaches as central proxy system **110***a* is not required to maintain and protect a translation table and cannot be forced to reveal the contents of any such table to a third party.

In an alternate advantageous embodiment, central proxy system **105***a* further includes a data store capable of containing e-mailboxes for the users and specific to the server sites. According to this embodiment, each user has a mailbox for each site that has generated mail destined for the user. Rather than compromising security by allowing automatic remailing to the user, the present embodiment may store e-mail for explicit retrieval by each user.

For each server, it may be advantageous for users to have a separate e-mail box, possibly identified by user-substitute identifiers. This approach may allow for suitable disposal of e-mail messages received from the third-parties (e.g., "junk e-mail") as well as the option of selective disposal of e-mail messages.

In one advantageous embodiment, each of e-mailboxes has a key associated therewith, the key being a function of the data and an index number. The use of keys with e-mailboxes is known. In another advantageous embodiment, central proxy system **110***a* further comprises a computer-executable routine that, given the substitute identifiers, collects e-mail destined for the users and contained within a plurality of site-specific e-mailboxes. This embodiment may suitably employ a mail-collecting routine that automatically locates user site **105***a*'s various mailboxes and retrieves the mail therefrom once the user has supplied the appropriate data.

According to one advantageous embodiment, central proxy system **110***a* includes functionality necessary to support electronic payment, the users employ electronic payment information to engage in anonymous commerce with the server sites. To facilitate the same, central proxy system **110***a* may include a data store capable of containing such electronic payment information. Further, substitute identifiers may be constructed, at least in part, using credit/debit card numbers, bank branch or account numbers, postal addresses, telephone numbers, tax identification numbers, social security numbers or the like. Various methods for achieving anonymous commerce are known.

By way of further example, an ever increasing number of sites require a valid credit card number as part of establishing an account, so that such sites may charge the user for their services (e.g., WALL STEET JOURNAL®, ESPN®, etc.). While the above-described proxy system provides substitute identifiers to free users from remembering these items and by providing a guard on (involuntary) data flowing to the web-site, it may not provide complete anonymity to a user who has provided a credit card number to a site. One solution, described briefly above, requires central proxy system **110***a* to provide its own valid credit card number to

the requesting site and then collect money from its users. If central proxy system **105***a* is incorporated into an Internet provider, for example, such as AMERICA ONLINE®, then this relationship may already exist.

Alternatively, central proxy system **110***a* may be known and trusted by other sites, thereby allowing central proxy system **110***a* to generate an alias credit card number and expiration date, and then to authenticate this data and send it to a requesting site. The site can then check that this number indeed originates from central proxy system **110***a* and hence accepts the same as valid, with the understanding that it can collect the money from central proxy system **110***a*. There no longer is a need to send a "real" credit card number between central proxy system **110***a* and the sites.

It is important to realize that the various features and aspects of the embodiments above-described may also be suitably implemented in accordance with the peripheral proxy system described with reference to FIG. **1**. More particularly, turning momentarily to FIG. **5**, there is illustrated a block diagram of an exemplary sub-network (generally designated **500**) of the distributed network of FIG. **1** showing a peripheral proxy system **120** that includes each of user site **105***a*, central proxy system **110***a* and NYT-site **110***g* (shown among a plurality of other illustrative server sites **110** of Internet 115) according to the principles of the present invention.

Peripheral proxy system **120**, as set forth above, includes first and second executable routines. The first routine, which advantageously resides in user site **105***a*, constructs substitute identifiers from data particular to user site **105***a*. The second routine, which also illustratively resides in user site **105***a*, transmits the substitute identifiers to central proxy system **110***a*. Central proxy system **110***a* then retransmits the substitute identifiers to server site **110***g* and thereafter communicates (e.g., transmits, receives, etc.) information (e.g., browsing commands, data, etc.) between user site **105***a* to server site **110***g*. This second configuration is particularly advantageous when users may not trust central proxy system **110***a* or the communication lines therebetween, and want to keep user identifications and other secret information secure.

A local proxy system **510** may be used to maintain the same, and may use the user's identification and other information to compute the substitute identifiers. Local proxy system **510** communicates with a central proxy system **110***a*, which may be used to forward communication to servers and handle e-mail. In one embodiment, central proxy system **110***a* communicates with computer-executable local routines associated with the users, the local routines constructing the site-specific substitute identifiers from data specific to the users. Again, central proxy system **110***a* may rely on distributed routines, local to each user, that generate the substitute identifiers and transmit the same to central proxy system **110***a*.

Turning now to FIG. **6**, illustrated is a block diagram of an exemplary sub-network (generally designated **600**) of the distributed network **100** including each of user site **105***a*, central proxy system **110***a* and a plurality of illustrative server sites **110***b*, **110***c*, and **110***g* according to an exemplary marker proxy embodiment of the present invention. As described above, the central proxy system of the present invention may be employed in at least two configurations, namely, a central proxy configuration (FIG. **2**) or a peripheral proxy configuration (FIG. **5**).

In the central proxy configuration, central proxy system **110***a* computes substitute identifiers. An implementation of this configuration may require user site **105***a* to provide one or more user definable character strings (e.g., user identification, password and other secret information) once, and central proxy system **110***a* will thereafter generate the substitute identifiers as needed. Central proxy system **110***a* may associate the user definable character strings with a series of HTTP requests generated by the same user site **105***a*—the central proxy system **110***a* may associate each request with a session, that contains all communication between a specific user site **105***a* and the central proxy system **110***a*.

The HTTP protocol however does not generally directly support sessions or relationships between requests. More particularly, each HTTP request may be sent a new socket connection, and there is no required HTTP header field that can link successive requests from the same user.

It should be noted that the session identification is typically not necessary in the peripheral proxy configuration since central proxy system **110***a* may forward communications without any computation. In a typical embodiment, peripheral proxy system **120** retransmits browsing commands received from user site **105***a* to central proxy system **110***a*, which then retransmits such commands to server site **110***g*. According to one embodiment, peripheral proxy system **120** removes and, possibly, substitutes portions of the browsing commands that would identify user site **105***a* to server site **110***g*.

In one advantageous embodiment user site **105***a* runs a marker program **605** locally. Marker program **605** operates to tag user site **105***a*'s requests with a session tag, t. Central proxy system **110***a* uses this tag to identify requests belonging to a particular one of a group of users. Marker program **605** may be implemented to store user site **105***a*'s session tag and add this tag to all requests, and central proxy system **110***a* removes the session tag before forwarding the request to some server site. The session tag should be unique, as no two users should have the same tag.

It should be noted that NETSCAPE® uses "cookies," which are a mechanism for storing and retrieving long term session information (the use of "cookies" conceptually is known). The cookies are generated by the browsed servers and are associated with a specific domain name. Browsers **300** submit the cookies associated with a specific domain name whenever the user re-visits that domain. Servers typically only generate cookies associated with their domain. Cookies provide an easy mechanism to keep session information, such as the contents of a "shopping cart," account name, password, event counters, user preferences, etc.

Some companies, use cookies extensively to track users and their habits. Since the proxy systems of the present invention present substitute identifiers to browsed servers, the servers cannot learn true user identities. Thus all of the information that the server may store in its cookie relates to some "alias persona," and not to the true user. Whenever the user returns to the same server, it will present the same substitute identifiers, and may also submit the cookie that the server generated earlier for this alias persona.

It is apparent from above, that the present invention provides, for use with a network having user sites and server sites, wherein the server sites are capable of being browsed by the user sites based on identifiers received into the server sites and personal to the user sites, both a central and a peripheral proxy system for providing consistent substitute identifiers to the server sites that allow the user sites to browse the server sites in an anonymous and personal fashion via the proxy system.

An exemplary central proxy system includes: (1) an executable first routine that processes site-specific substitute

identifiers constructed from data specific to the user sites, (2) an executable second routine that transmits the substitute identifiers to the server sites and thereafter retransmits browsing commands received from the user sites to the server sites and (3) an executable third routine that removes (and possibly substitutes) portions of the browsing commands that would identify the user sites to the server sites.

An exemplary peripheral proxy system includes: (1) an executable first routine that constructs a particular substitute identifier from data received from a particular user site and (2) an executable second routine that transmits the particular substitute identifier to a central proxy system, the central proxy system then retransmitting the particular substitute identifier to the server site and thereafter retransmitting browsing commands received from the particular user site to the server site.

Although the present invention has been described in detail, those skilled in the art should understand that they can make various changes, substitutions and alterations herein without departing from the spirit and scope of the invention in its broadest form. More particularly, it should be apparent to those skilled in the pertinent art that the above-described routines are software-based and executable by a suitable conventional computer system/network. Alternate embodiments of the present invention may also be suitably implemented, at least in part, in firmware or hardware, or some suitable combination of at least two of the three. Such firmware- or hardware embodiments may include multi, parallel and distributed processing environments or configurations, as well as alternate programmable logic devices, such as programmable array logic ("PALs") and programmable logic arrays ("PLAs"), digital signal processors ("DSPs"), field programmable gate arrays ("FPGAs"), application specific integrated circuits ("ASICs"), large scale integrated circuits ("LSIs"), very large scale integrated circuits ("VLSIs") or the like—to form the various types of modules, circuitry, controllers, routines and systems described and claimed herein.

Conventional computer system architecture is more fully discussed in *The Indispensable PC Hardware Book,* by Hans-Peter Messmer, Addison Wesley (2nd ed. 1995) and *Computer Organization and Architecture,* by William Stallings, MacMillan Publishing Co. (3rd ed. 1993); conventional computer, or communications, network design is more fully discussed in *Data Network Design,* by Darren L. Spohn, McGraw-Hill, Inc. (1993); and conventional data communications is more fully discussed in *Voice and Data Communications Handbook,* by Bud Bates and Donald Gregory, McGraw-Hill, Inc. (1996), *Data Communications Principles,* by R. D. Gitlin, J. F. Hayes and S. B. Weinstein, Plenum Press (1992) and *The Irwin Handbook of Telecommunications,* by James Harry Green, Irwin Professional Publishing (2nd ed. 1992). Each of the foregoing publications is incorporated herein by reference for all purposes.

What is claimed is:

1. A central proxy system for coupling to a network and for allowing users to browse server sites on said network anonymously via said central proxy system, said central proxy system comprising:

a computer-executable first routine that processes site-specific substitute identifiers constructed from non-masked data specific to said users, such that said server sites are unable to determine an identity of said user;

a computer-executable second routine that transmits said substitute identifiers to said server sites and thereafter retransmits browsing commands received from said users to said server sites; and

a computer-executable third routine that removes portions of said browsing commands that would identify said users to said server sites.

2. The central proxy system as recited in claim 1 wherein said data comprises identification data and a user definable character string supplied by said users.

3. The central proxy system as recited in claim 1 wherein said site-specific substitute identifiers comprise site-specific substitute user names and site-specific substitute user passwords.

4. The central proxy system as recited in claim 1 wherein said first routine constructs site-specific substitute electronic mail addresses for said users from said data.

5. The central proxy system as recited in claim 1 wherein said first routine constructs said site-specific substitute identifiers from addresses of said server sites.

6. The central proxy system as recited in claim 1 wherein said server sites are World Wide Web sites capable of presenting web pages to said users, said second routine transmitting said substitute identifiers to said server sites under direction of said users.

7. The central proxy system as recited in claim 1 wherein said second routine transmits said substitute identifiers to said server sites based on alphanumeric codes supplied in web page fields by said users.

8. The central proxy system as recited in claim 7 wherein said alphanumeric codes are arranged in escape sequences.

9. The central proxy system as recited in claim 7 wherein said users manually place said alphanumeric codes in said web page fields.

10. The central proxy system as recited in claim 9 wherein said central proxy system communicates with computer-executable local routines associated with said users, said local routines constructing said site-specific substitute identifiers from data specific to said users.

11. The central proxy system as recited in claim 1 further comprising a data store capable of containing electronic mail destined for said users.

12. The central proxy system as recited in claim 1 wherein said first routine processes substitute identifiers constructed by applying pseudo-random and hash functions to said data received from said users.

13. The central proxy system as recited in claim 1 further comprising a data store capable of containing electronic mailboxes for said users and specific to said server sites.

14. The central proxy system as recited in claim 13 wherein each of said electronic mailboxes has a key associated therewith, said key being a function of said data and an index number.

15. The central proxy system as recited in claim 1 further comprising a computer-executable routine that, given said substitute identifiers, collects electronic mail destined for said users and contained within a plurality of site-specific electronic mailboxes.

16. The central proxy system as recited in claim 1 wherein said first routine receives session tags added to said browsing commands, said central proxy system employing said session tags to associate said substitute identifiers with each of said browsing commands.

17. The central proxy system as recited in claim 1 further comprising a data store capable of containing session information specific to said users and accessible by said server sites.

18. The central proxy system as recited in claim 1 further comprising a data store capable of containing electronic payment information, said users employing said electronic payment information to engage in anonymous commerce with said server sites.

19. The central proxy system as recited in claim 1 further comprising an initializing routine that constructs said site-specific substitute identifiers from data specific to said users and communicates said site-specific substitute identifiers to said first routine.

20. A peripheral proxy system for coupling to a network and for allowing at least one user to browse a server site on said network anonymously via a central proxy system, said peripheral proxy system comprising:

a computer-executable first routine that constructs a particular substitute identifier from non-masked data received from a particular user, such that said server site is unable to determine an identity of said user; and

a computer-executable second routine that transmits said particular substitute identifier to said central proxy system, said central proxy system retransmitting said particular substitute identifier to said server site and thereafter retransmitting browsing commands received from said particular user to said server site.

21. The peripheral proxy system as recited in claim 20 wherein said data comprises identification data and a user definable character string supplied by said particular user.

22. The peripheral proxy system as recited in claim 20 wherein said particular substitute identifier comprises a particular substitute user name and a particular substitute user password.

23. The peripheral proxy system as recited in claim 20 wherein said first routine constructs a particular substitute electronic mail address for said particular user from said data.

24. The peripheral proxy system as recited in claim 20 wherein said first routine constructs said particular substitute identifier from an address of said server site, said particular substitute identifier therefore being specific to said server site.

25. The peripheral proxy system as recited in claim 20 wherein said server site is a World Wide Web site capable of presenting at least one web page to said users, said central proxy system transmitting said particular substitute identifier to said server site under direction of said particular user.

26. The peripheral proxy system as recited in claim 20 wherein said central proxy system said particular substitute identifier to said server site based on alphanumeric codes supplied in web page fields by said user.

27. The peripheral proxy system as recited in claim 26 wherein said alphanumeric codes are arranged in escape sequences.

28. The peripheral proxy system as recited in claim 20 wherein said central proxy system further comprises a computer-executable third routine that removes portions of said browsing commands that would identify said particular user to said server site.

29. The peripheral proxy system as recited in claim 28 wherein said first and second routines are executable on a computer system associated with said particular user and said central proxy system is a computer system having a network address different from said computer system associated with said particular user.

30. The peripheral proxy system as recited in claim 20 wherein said central proxy system further comprises a data store capable of containing electronic mail destined for said particular user.

31. The peripheral proxy system as recited in claim 20 wherein said first routine constructs said particular substitute identifier by applying pseudo-random and hash functions to said data received from said particular user.

32. The peripheral proxy system as recited in claim 20 wherein said central proxy system further comprises a data store capable of containing an electronic mailbox for said particular user and specific to said server site.

33. The peripheral proxy system as recited in claim 32 wherein said electronic mailbox has a key associated therewith, said key being a function of said data and an index number.

34. The peripheral proxy system as recited in claim 20 wherein said central proxy system further comprises a computer-executable routine that, given said particular substitute identifier, collects electronic mail destined for said particular user and contained within at least two electronic mailboxes.

35. The peripheral proxy system as recited in claim 20 wherein said central proxy system further comprises a computer-executable marker routine that adds session tags to said browsing commands, said proxy system employing said session tags to associate said particular substitute identifier with each of said browsing commands.

36. The peripheral proxy system as recited in claim 20 wherein said central proxy system further comprises a data store capable of containing session information specific to said particular user and accessible by said server site.

37. The peripheral proxy system as recited in claim 20 wherein said central proxy system further comprises a data store capable of containing electronic payment information, said particular user employing said electronic payment information to engage in anonymous commerce with said server site.

38. A method for use with a network having a server site capable of being browsed by users and for allowing said users to browse said server site on said network anonymously via said proxy system, said method comprising the steps of:

constructing a particular substitute identifier from non-masked data received from a particular user, such that said server site is unable to determine an identity of said user;

transmitting said particular substitute identifier to said server site; and

thereafter retransmitting browsing commands received from said particular user to said server site.

39. The method as recited in claim 38 wherein said data comprises identification data and a user definable character string supplied by said particular user.

40. The method as recited in claim 38 wherein said particular substitute identifier comprises a particular substitute user name and a particular substitute user password.

41. The method as recited in claim 38 further comprising the step of constructing a particular substitute electronic mail address for said particular user from said data.

42. The method as recited in claim 38 wherein said step of constructing comprises the step of constructing said particular substitute identifier from an address of said server site, said particular substitute identifier therefore being specific to said server site.

43. The method as recited in claim 38 wherein said server site is a World Wide Web site capable of presenting at least one web page to said users, said method further comprising the step of transmitting said particular substitute identifier to said server site under direction of said particular user.

44. The method as recited in claim 38 wherein said step of transmitting comprises the step of transmitting said particular substitute identifier to said server site based on alphanumeric codes supplied in web page fields by said user.

45. The method as recited in claim 44 wherein said alphanumeric codes are arranged in escape sequences.

46. The method as recited in claim 38 further comprising the step of removing portions of said browsing commands that would identify said particular user to said server site.

47. The method as recited in claim 46 wherein said step of constructing is performed on a computer system associated with said particular user and said steps of transmitting and thereafter transmitting are performed on a computer system having a network address different from said computer system associated with said particular user.

48. The method as recited in claim 38 further comprising the step of storing electronic mail destined for said particular user.

49. The method as recited in claim 38 wherein said step of constructing comprises the step of applying pseudo-random and hash functions to said data received from said particular user.

50. The method as recited in claim 38 further comprising the step of creating an electronic mailbox for said particular user and specific to said server site.

51. The method as recited in claim 50 wherein said electronic mailbox has a key associated therewith, said key being a function of said data and an index number.

52. The method as recited in claim 38 further comprising the step of collecting electronic mail destined for said particular user and contained within at least two electronic mailboxes given said particular substitute identifier.

53. The method as recited in claim 38 further comprising the step of adding session tags to said browsing commands, said proxy system employing said session tags to associate said particular substitute identifier with each of said browsing commands.

54. The method as recited in claim 38 further comprising the step of storing session information specific to said particular user and accessible by said server site.

55. The method as recited in claim 38 further comprising the step of storing electronic payment information, said particular user employing said electronic payment information to engage in anonymous commerce with said server site.

* * * * *

US006208991B1

(12) **United States Patent**
French et al.

(10) **Patent No.: US 6,208,991 B1**
(45) **Date of Patent: Mar. 27, 2001**

(54) **DYNAMIC FILE MAPPING FOR NETWORK COMPUTERS**

(75) Inventors: **Steven Michael French**; **Chakkalamattam Jos Paul**, both of Austin; **James Richard Schoech**, Round Rock, all of TX (US)

(73) Assignee: **International Business Machines Corporation**, Armonk, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 0 days.

(21) Appl. No.: **09/140,169**

(22) Filed: **Aug. 26, 1998**

(51) **Int. Cl.**[7] **G06F 15/173**

(52) **U.S. Cl.** **707/10**; 707/9; 709/203; 709/219; 709/225; 709/229; 713/201; 713/202

(58) **Field of Search** 707/10, 9, 201; 709/203, 225, 219, 229, 239

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,732,214	*	3/1998	Subrahmanyam	709/227
5,796,952	*	8/1998	Davis et al.	709/224
5,875,327	*	2/1999	Brandt et al.	395/651
5,926,637	*	7/1999	Cline et al.	395/701
5,974,572	*	10/1999	Weinberg et al.	714/47
6,018,745	*	1/2000	Kuftedjian	707/200
6,029,168	*	2/2000	Frey	707/10
6,061,795	*	5/2000	Dircks et al.	713/201
6,070,242	*	5/2000	Wong et al.	713/201

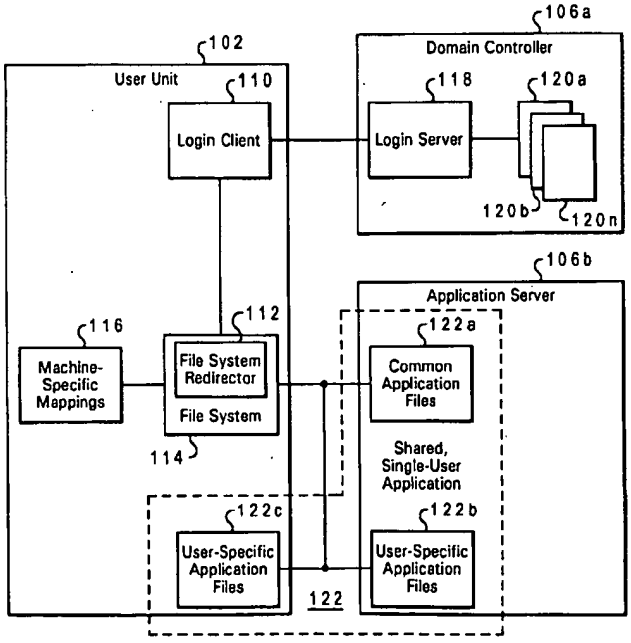
* cited by examiner

Primary Examiner—Jean R. Homere
(74) *Attorney, Agent, or Firm*—Jeffrey S. LaBaw; Felsman, Bradley, Vaden, Gunter & Dillon, LLP

(57) **ABSTRACT**

File mappings are dynamically loaded within an operating data processing system appended to existing file or directory mappings during operation rather than being loaded during system initialization. A triggering event, such as a user logging into a network from the data processing system with a unique userid, initiates the process for selectively loaded the dynamic file mappings. A context variable, such as the userid of the user logging into the network, is employed to select the set or table of file mappings which are dynamically loaded. The dynamically loaded file mappings are appended to traditional, machine-specific file mappings loaded at system initialization and may be unloaded without affecting such traditional file mappings. The capability of dynamic file mapping allows single-user applications, those designed for use by only one user at a time, to be shared from a single network location, with user-specific files mapped to different locations for different users. Thus, only one copy of a browser is required in a network computer environment. Sensitive files, such as bookmark, security, and cookie files, are dynamically mapped to a user-specific directory based on the userid regardless of the user unit from which the user logs in. Multiple users may share a single copy of the single user browser, and users may "roam" the network, logging in at any data processing system.

10 Claims, 4 Drawing Sheets



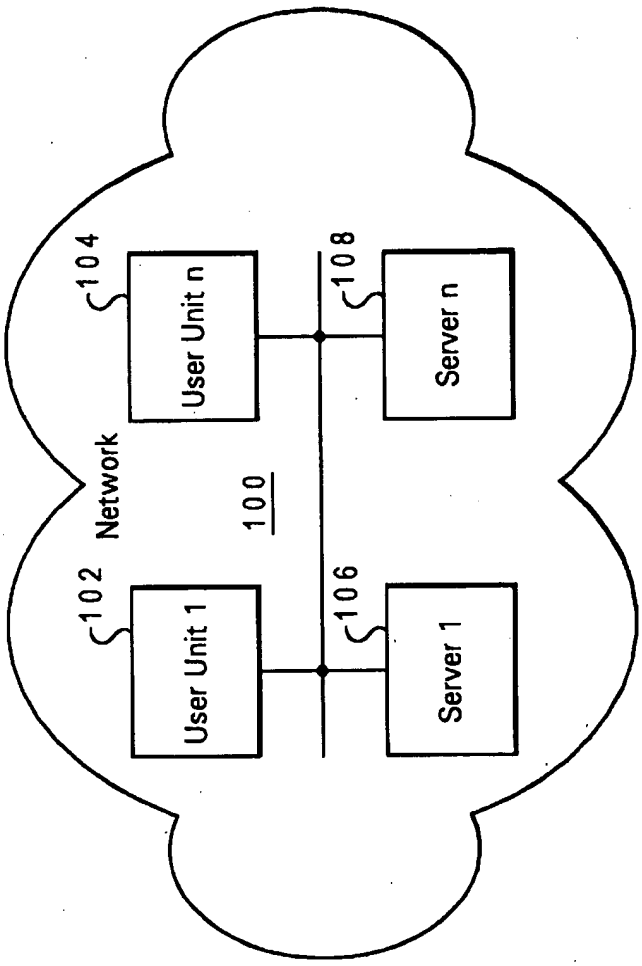

Network
100
User Unit 1
102
User Unit n
104
Server 1
106
Server n
108

Fig. 1A

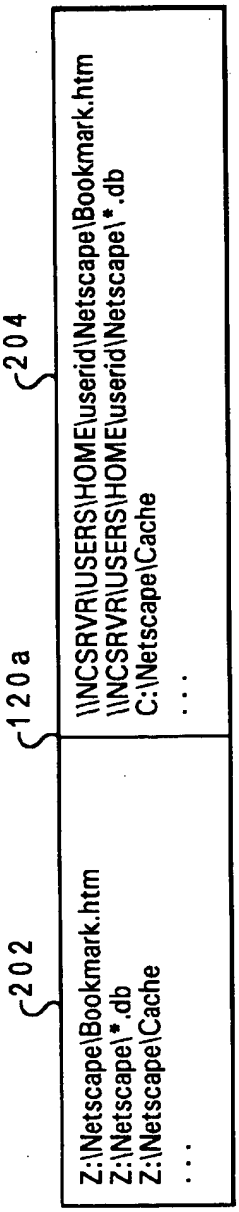

202
Z:\Netscape\Bookmark.htm
Z:\Netscape*.db
Z:\Netscape\Cache
...
120a
\\NCSRVR\USERS\HOME\userid\Netscape\Bookmark.htm
\\NCSRVR\USERS\HOME\userid\Netscape*.db
C:\Netscape\Cache
...
204

Fig. 2

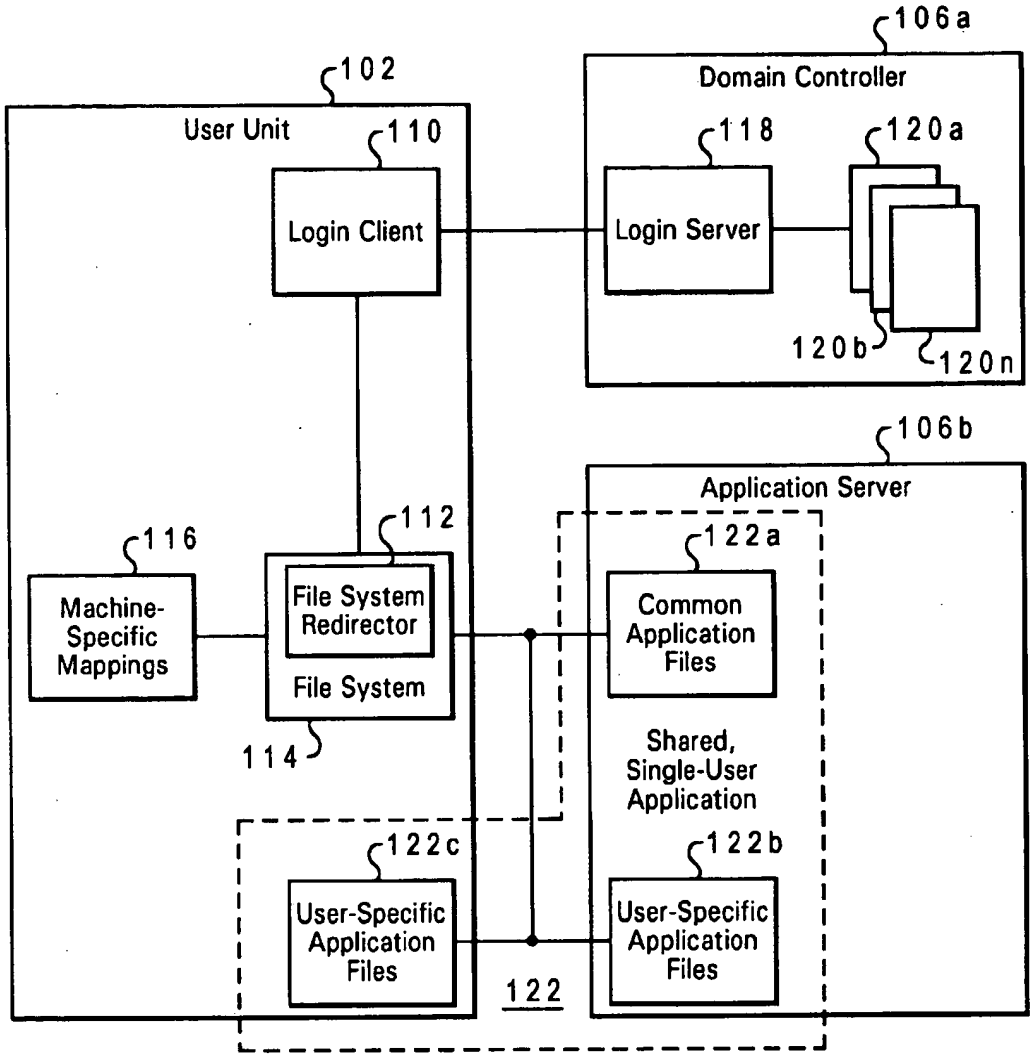

106a
Domain Controller
102
User Unit
110
118
120a
Login Client
Login Server
120b
120n
106b
Application Server
112
116
122a
File System Redirector
Machine-Specific Mappings
Common Application Files
File System
114
Shared, Single-User Application
122c
122b
User-Specific Application Files
User-Specific Application Files
122

Fig. 1B

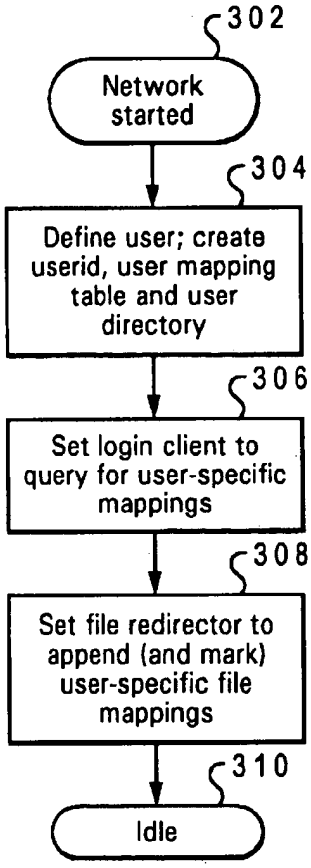

302
Network started
304
Define user; create userid, user mapping table and user directory
306
Set login client to query for user-specific mappings
308
Set file redirector to append (and mark) user-specific file mappings
310
Idle

Fig. 3A

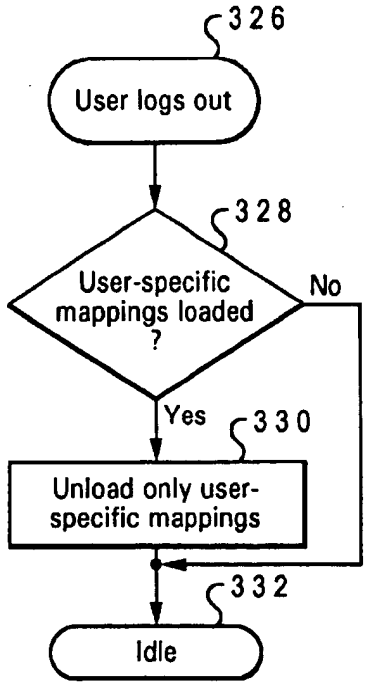

326
User logs out
328
User-specific mappings loaded ?
No
Yes
330
Unload only user-specific mappings
332
Idle

Fig. 3C

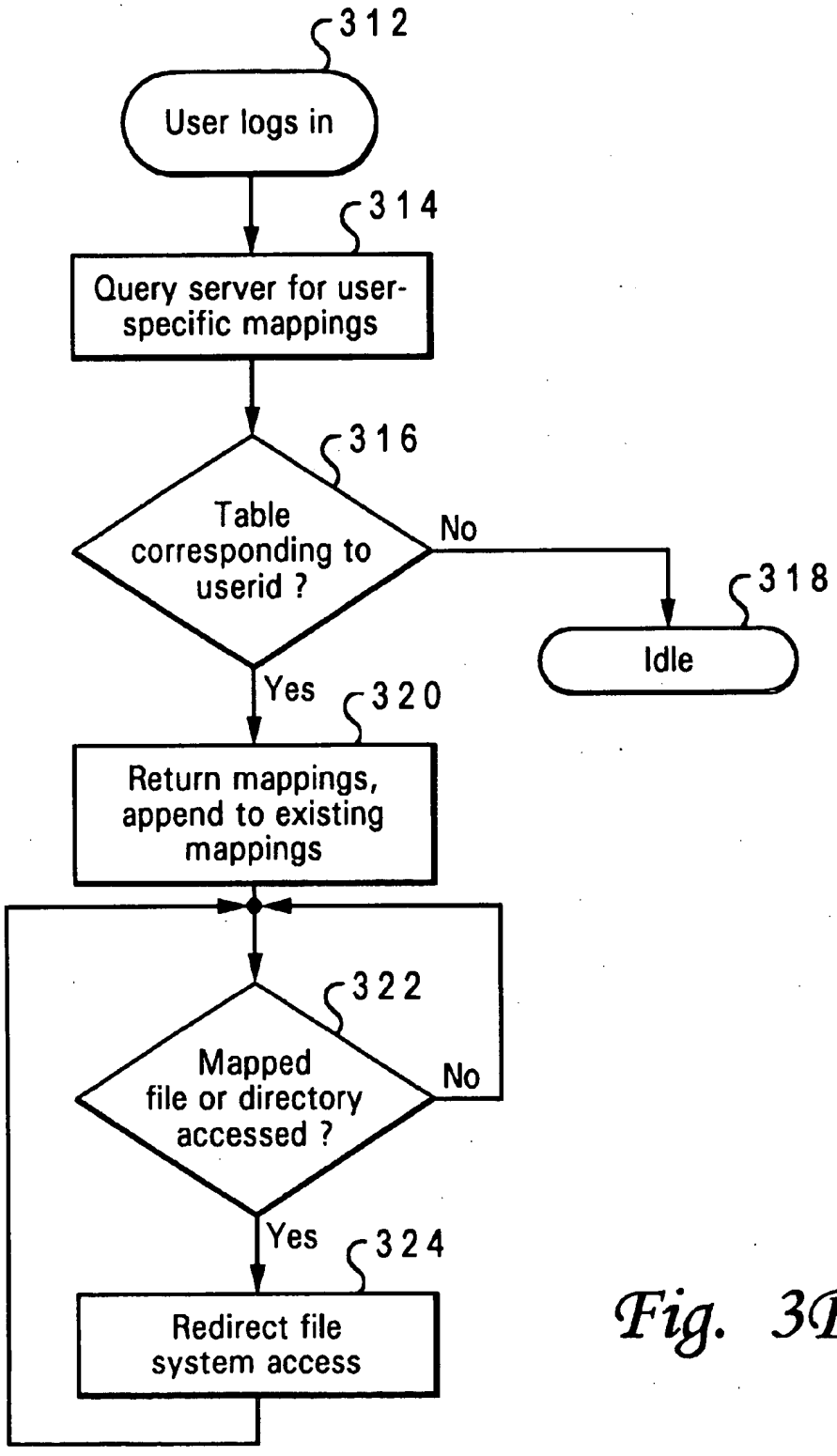

312
User logs in
314
Query server for user-specific mappings
316
Table corresponding to userid ?
No
318
Idle
Yes
320
Return mappings, append to existing mappings
322
Mapped file or directory accessed ?
No
Yes
324
Redirect file system access

Fig. 3B

DYNAMIC FILE MAPPING FOR NETWORK COMPUTERS

BACKGROUND OF THE INVENTION

1. Technical Field

The present invention relates in general to application file management within data processing systems and in particular to application- and user-specific file mapping within data processing systems or networks. Still more particularly, the present invention relates to user-based dynamic file mapping allowing sharing of single-user applications.

2. Description of the Related Art

Applications may be stored on network servers and run from workstations connected to the network. Different users authorized access to the network may run the application remotely from different workstations connected to the network. In the best cases, however, significant duplication of files is typically required. In some cases, a complete copy of the application for each user must be stored on the server.

Currently, the only mechanisms available for transparently remapping application files are machine-specific. Unix's symbolic links, for example, allow a dummy filename to be mapped to a secondary location, but only in a machine-specific fashion. PC RIPL (Remote Boot) FileIndirectionTables are also machine-specific and are loaded at boot time. Distributed File System (DFS) implements "referrals," which provide static directory mapping, which is constant across all machines and users.

Machine-specific file remappings typically cannot handle the case of configuration files. Single user applications—i.e. applications which are not designed for concurrent use by multiple users—include configuration files containing user preferences, cache or data files storage locations, etc. Such configuration files are user specific and typically should be stored in different, access-restricted directories.

It would be desirable, therefore, to provide a mechanism for dynamically remapping files associated with single user applications in a machine-independent fashion.

SUMMARY OF THE INVENTION

It is therefore one object of the present invention to provide an improved method and apparatus for application file management within data processing systems.

It is another object of the present invention to provide an improved method and apparatus for application- and user-specific file mapping within data processing systems or networks.

It in yet another object of the present invention to provide a method and apparatus for user-based dynamic file mapping allowing sharing of single-user applications.

The foregoing objects are achieved as is now described. File mappings are dynamically loaded within an operating data processing system appended to existing file or directory mappings during operation rather than being loaded during system initialization. A triggering event, such as a user logging into a network from the data processing system with a unique userid, initiates the process for selectively loading the dynamic file mappings or reloading new dynamic file mappings. A context variable, such as the userid of the user logging into the network, is employed to select the set or table of file mappings which are dynamically loaded. The dynamically loaded file mappings are appended to traditional, machine-specific file mappings loaded at system initialization and may be unloaded without affecting such traditional file mappings. The capability of dynamic file mapping allows single-user applications, those designed for use by only one user at a time, to be shared from a single network location, with user-specific files mapped to different locations for different users. Thus, only one copy of a browser is required in a network computer environment. Sensitive files, such as bookmark, security, and cookie files, are dynamically mapped to a user-specific directory based on the userid regardless of the user unit from which the user logs in. Multiple users may share a single copy of the single user browser, and users may "roam" the network, logging in at any data processing system.

The above as well as additional objects, features, and advantages of the present invention will become apparent in the following detailed written description.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself however, as well as a preferred mode of use, further objects and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

FIGS. **1A** and **1B** depict physical and functional block diagrams of a data processing system network in accordance with a preferred embodiment of the present invention;

FIG. **2** is a user-specific mapping table for dynamic file mapping in accordance with a preferred embodiment of the present invention;

FIGS. **3A** and **3B** depict high level flowcharts for processes of establishing and employing dynamic file mapping in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

With reference now to the figures, and in particular with reference to FIGS. **1A–1B**, physical and functional block diagrams of a data processing system network in accordance with a preferred embodiment of the present invention is depicted. FIG. **1A** depicts the physical components of network **100**, which is preferably a local area network (LAN) of one of various types known in the art (Ethernet, etc.). Network **100** in the exemplary embodiment includes a plurality of user units **102** and **104** and a plurality of servers **106** and **108**. The present invention may be employed with only a single user unit and a single server, and the user unit and the server unit may be colocated. However, the benefits of the present invention may best be obtained within a network including a plurality of user units and at least one server. User units **102** and **104** and servers **106** and **108** are all data processing systems interconnected by a network communications systems in accordance with the known art.

FIG. **1B** depicts a functional block diagram of the network components comprising the present invention. Each user unit within network **100** supporting dynamic file remapping in accordance with the present invention, such as user unit **102** in the depicted example, includes a login client application **110** and a file system redirector **112**. File system redirector **112**, which preferably forms part of the file system **114**, may be integrated into the operating system (together with the file system) of user unit **102** in accordance with the known art. User unit **102** may optionally include a machine-specific set **116** of file or directory mappings as known in the art, such as the mappings which are utilized to permit access to a shared disk drive within network **102**.

Server **106** includes a login server **118** and one or more sets **120***a*–**120***n* of dynamic or user-specific file/directory mappings. User-specific sets of file and/or directory mappings **120***a*–**120***n* may be of the same type and form as conventional machine-specific file mappings **116** known in the art, and may be loaded or employed by file redirector **112** and file system **114** in the same manner. In the present invention, however, the cause for loading and setting the additional file mappings and the selection of a particular set of file mappings differs from the prior art, as described in further detail below. Server **106** also includes a shared, single-user application **122**. Single user application **122** may be any conventional application designed for use by only one user at a time, as opposed to multi-user applications which allow multiple users to concurrently share the application. Although the exemplary embodiment depicts login server **118** and single user application **122** as stored within a single server **106**, these components may be stored on separate servers within the network. In fact, different portions of single user application **122** may be stored on separate servers as described in further detail below.

The client or user unit **102** may obtain four items from one or more server units **106** within a network: (a) the application program; (b) the application data, which may be user-specific and need not be colocated on the server unit(s) for the application program; (c) user-based file mappings; and (d) machine-based file mappings. The user-based file mappings are preferably retrieved from a central logon server (more commonly referred to as a "domain controller") or its backup logon server. These user-based file mappings may be employed to transparently locate the application data for a particular user. The machine-based (or "machine-specific") file mappings are optional and may be obtained either locally or from a server unit, but preferably from a network boot server.

While the exemplary embodiment suggests that the application program, application data, and user-based file mappings are colocated in the same server unit **106**, those skilled in the art will recognize that, in practice, the application program and the user-based (or "user-specific") file mappings would rarely be colocated on the same server unit.

In the present invention, login client **110** is employed to establish a functional connection between user unit **102** and network **100**, submitting a request to login server **118** together with a userid and password as is known in the art. User-specific file mapping tables **120***a*–**120***n* are stored at a location within server **106** known to the domain controller within logon authentication server application **118**, When a user employs logon client **110** to connect to server **106**, the client logon code within user unit **102** queries server **106** for user-specific mapping information. Server **106** determines whether a user-specific mapping table within tables **120***a*–**120***n* corresponds to the logon id supplied by the user logging on. If so, the appropriate user-specific file mappings are returned to user unit **102** by login server **118**. In an alternative implementation of the present invention, the login client **110** would retrieve the user-specific mapping information directly (e.g. using redirected file i/o) from a well known, user-specific location on the domain controller or backup domain controller.

User-specific file mappings returned by the logon authentication server application **118** are established for user unit **102**, appended to the currently active file mapping information in the client-side file system redirector **112** within user unit **102**. These user-specific file mappings remain in effect and are employed by the file system redirector **112** within user unit **102** until the user logs off that data processing system. When the user logs off a particular user unit, the user-specific file mappings are unloaded without affecting other remapping information within user-unit **102**, such as machine specific file mappings **116** loaded at system initialization to establish a connection to network **100** via a network interface card for submitting login requests. Another set of file mappings may than be loaded for the next user that logs on to network **100** from user unit **102** with a different userid.

An example of a suitable single user application **122** is Netscape Navigator version 2 available from Netscape Communications Corporation of Mountain View, Calif. (Netscape Communicator, which is version 4 of the Netscape browser application, is a multi-user application). The present invention may be utilized for sharing a single-user Netscape Navigator application among various users in a network computer (NC) environment.

In the prior art, in order to support user-specific preferences, bookmarks, and caching, a unique copy of Netscape Navigator would be required on each user-unit employed by only a single user or in each user's home directory on the network. With the present invention, a single copy of most of the Netscape Navigator application's files are stored in a single shared server directory. Some files, such as the Web page cache, are remapped to a machine specific directory while the remaining files, such as the bookmarks file (BOOKMARK.HTM), the Netscape configuration file (NETSCAPE.INI), security files and the cookie cache, are dynamically mapped to another location depending on the identity of the user logged on. This protects the user's sensitive security and cookie data and user preferences even as if the user "roams" from user unit to user unit within the network or if multiple users share the same user unit.

As illustrated by the example described above, the dynamic file remapping of the present invention allows the application files of single-user application **122** to effectively be distributed among a plurality of data processing systems, or within different directories, depending on the user. A single copy of common application files **122***a*, such as the executables, may be stored in a single directory and shared among a plurality of users. User-specific application files **122***b*, such as security-sensitive bookmark and cookie files, may be stored in a different location, with different sets of user-specific application files **122***b* mapping to the directory and file designation for those files within single user application **122**, depending upon the identity of the user. Finally, non-sensitive user-specific application files **122***c*, such as cached Web pages, may be stored in yet another location, including within either user unit **102** or server **106**. User specific files may be mapped to different locations for security or reliability purposes as well. For example, the netscape bookmark file may be mapped to a read-only directory with other read-only configuration files for this user. These read-only files may be more easily replicated from a common location, and the user will be unable to accidently corrupt these files through user error. Similarly, files such as the Netscape security files such as cookie files can be mapped to a read-write location. User specific files may be transparently mapped to common locations for groups of users as well (usually for read only configuration files). File or directory mappings for all such application files forming part of a conventional, single-user application may be dynamically specified based on the userid of the user.

Although the dynamic file mapping technique of the present invention provides greater benefit in conjunction

with single-user applications, value may still be provided from use with multi-user applications. For instance, dynamic file mapping provides greater ease of use (since user specific files can more easily be cloned and these files can even be mapped to common locations across a set of similar users of the applications. In addition, dynamic file mapping can provide significant savings in disk space even for multiuser applications since only the minimum set of non-shared files need be remapped to user specific areas, and some can be shared by classes of users of the application. Accordingly, the present invention may also be employed with more recent versions of the Netscape browser, and other multi-user applications.

Referring now to FIG. **2**, a user-specific mapping table for dynamic file mapping in accordance with a preferred embodiment of the present invention is illustrated. File mapping table **120***a* includes two columns. The first column **202** contains the client-side view of source directory or path of the files employed by the shared, single-user application. At least three types of source paths may be specified: explicit directory paths and file names, such as Z:\Netscape\Bookmark.htm; directory paths and wildcards, such as Z:\Netscape*.db; and directory names, such as Z:\Netscape\Cache.

The second column **204** of user-specific mapping table **120***a* contains the corresponding location of the files specified by name or directory path in column **202**. The files may be located on either a server or the user unit. When the client file system requests a file within a mapped directory path and/or filename, the requested file is obtained from the user-specific location on the server rather than from the drive specified. Each user will have a different directory, preferably based on the specific userid of that user, such as a directory path including the userid.

For a specific subset of users, a file can be mapped to the same location, by specifying the same target location in the file mapping table. For example, half of the users in the organization could share the same Netscape bookmark file from a read-only location on the network, and the other half of the users each could have their own read-write copy stored in a user specific location.

With reference now to FIGS. **3A–3B**, high level flowcharts for processes of establishing and employing dynamic file mapping in accordance with a preferred embodiment of the present invention are depicted. FIG. **3A** depicts a process of establishing dynamic file mapping. The process begins at step **302**, which depicts a network being formed with at least one client data processing system, or user unit, and at least one server data processing system, with a login client and server for providing functional connection of the user unit to the server.

The process next passes to step **304**, which illustrates defining a user for the network, including creating a userid for the user, creating a user-specific file mapping table at a location known to the login server, and user-specific directorial, if necessary. The process then passes to step **306**, which depicts setting the login client on the user unit to query for user-specific file mappings during the login routine. The process next passes to step **308**, which illustrates setting the file redirector to append or otherwise integrate user-specific file mappings to existing file mappings in the client data processing system network filesystem. The process then passes to step **310**, which depicts the process becoming idle until the invention is to be implemented in another network.

FIG. **3B** depicts a process of setting and employing dynamic file mappings in accordance with a preferred embodiment of the present invention. The process begins at step **312**, which illustrates a user logging into the network at the client data processing system utilizing a userid and password in accordance with the known art. The process next passes to step **314**, which illustrates the login client querying the login server for user-specific file mappings, and then to step **316**, which depicts a determination by the login server of whether a table of user-specific file mappings corresponding to the provided userid exists.

Although the exemplary embodiment depicts the attempt to dynamically load file mappings as being triggered by a user logging in, other triggering events may be employed instead. For example, the start of an application employing dynamically mapped files may be employed as a triggering event. Furthermore, the userid is only one possible context variable utilized to select particular file mappings to be loaded. Other context variables may be utilized in addition to or in lieu of the userid for selecting a particular set of file mappings to be loaded, such as the data processing system id, an application identifier or type, and the like. The present invention applies equally regardless of triggering event or context variable utilized to automatically initiate dynamic file mapping and to select particular file mapping sets, provided that rebooting the client system is not required to integrate the dynamic file mappings within the filesystem redirector and that the dynamic file mappings are loaded and unloaded without affecting traditional or machine-based file mappings.

If no table of user-specific file mappings is found, the process proceeds to step **318**, which depicts the process becoming idle until another user logs in to the network. If a corresponding table is found, however, the process proceeds instead to step **320**, which illustrates the server returning the file mappings to the user unit, which appends them to existing file mappings within the user unit without disturbing the existing file mappings.

The process then passes to step **322**, which depicts a determination by the file system redirector of whether a dynamically mapped file or directory has been accessed. If not, the process returns to step **322** and continues polling for attempts to access a dynamically mapped file or directory. If an attempt to access the dynamically mapped files or directories is made, the process proceeds instead to step **324**, which illustrates redirecting the attempted file system access to the dynamically mapped location. The portion of the process depicted in steps **322** and **324** continues until interrupted or killed by an external process.

FIG. **3C** depicts a process of unloading dynamically loaded file mappings in accordance with the present invention. The process begins at step **326**, which depicts to occurrence of a triggering event such as the user logging out of the network. The process then passes to step **328**, which illustrates a determination of whether any dynamically loaded file mappings are presently loaded. Some means for distinguishing dynamically loaded file mappings from traditional, machine-specific file mappings, such as a flag or attribute associated with dynamically loaded file mappings, is required for this purpose.

If at least one dynamically loaded file mapping is identified, the process proceeds to step **330**, which depicts unloading only the dynamic file mappings, without disturbing other file mappings, and the process then proceeds to step **332**. If no dynamic file mappings are loaded, however, the process proceeds instead directly to step **332**, which illustrates the process becoming idle until another triggering event for unloading dynamic file and directory mappings is detected.

The present invention adds the capability of user-based file re-mappings which may be loaded at any time. The mappings are transparently integrated with traditional (machine-based) file mappings by the client network file-system. This provides the necessary support for allowing a single-user application to be shared by multiple users while providing user-specific file support and protecting the security of such user-specific files. The file mappings established by the present invention are dynamic, and may be loaded or unloaded and replaced without reinitializing the client data processing system by logging in and logging out with different userids.

It is important to note that while the present invention has been described in the context of a fully functional data processing system or network, those skilled in the art will appreciate that the mechanism of the present invention is capable of being distributed in the form of a computer readable medium of instructions in a variety of forms, and that the present invention applies equally regardless of the particular type of signal bearing media used to actually carry out the distribution. Examples of computer readable media include: nonvolatile, hard-coded type media such as read only memories (ROMs) or erasable, electrically programmable read only memories (EEPROMs), recordable type media such as floppy disks, hard disk drives and CD-ROMs, and transmission type media such an digital and analog communication links.

While the invention has been particularly shown and described with reference to a preferred embodiment, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention.

What is claimed is:

1. A method of dynamic file mapping, comprising:

storing an application on a first server and a plurality of sets of user-specific file mappings on a second server within a network;

responsive to a user logging into the network, determining a userid for the user;

determining whether a set of user-specific file mappings is associated with the userid;

responsive to determining that a set of user-specific file mappings is associated with the userid, loading the set of user-specific file mappings;

responsive to the user executing the application, utilizing the user-specific file mappings to redirect file calls from the application.

2. The method of claim **1**, wherein the step of storing an application on a first server and a plurality of sets of user-specific file mappings on a second server within a network further comprises:

storing a plurality of sets of user-specific file mappings for the application on the second server.

3. The method of claim **1**, wherein the step of storing an application on a first server and a plurality of sets of user-specific file mappings on a second server within a network further comprises:

storing a single-user application on the first server for execution by a plurality of users.

4. The method of claim **1**, further comprising:

loading at least one machine-specific file mapping in addition to the set of user-specific file mappings.

5. The method of claim **1**, further comprising:

responsive to the user logging out of the network, unloading only the set of user-specific file mappings.

6. A mechanism for dynamic file mapping, comprising:

an application server containing an application;

a domain controller containing a plurality of sets of user-specific file mappings;

a user unit connected to the application server and the domain controller to form a network, wherein the network further comprises a logon routine:

determining a userid for a user in response to the user logging into the network;

determining whether a set of user-specific file mappings is associated with the userid; and

loading the set of user-specific file mappings in the user unit in response to determining that a set of user-specific file mappings is associated with the userid,

wherein the user unit, responsive to the user executing the application, utilizes the user-specific file mappings to redirect file calls from the application.

7. The mechanism of claim **6**, wherein the sets of user-specific file mappings further comprise:

user-specific file mappings for the application.

8. The mechanism of claim **6**, wherein the application further comprises:

a single-user application accessible for execution by a plurality of users.

9. The mechanism of claim **6**, wherein the user unit loads at least one machine-specific file mapping in addition to the set of user-specific file mappings.

10. The mechanism of claim **6**, wherein the network further comprises:

a logout routine unloading only the set of user-specific file mappings from the user unit in response to the user logging out of the network.

* * * * *